



Ana Margarida Varela Sebastião

Licenciada em Engenharia Electrotécnica e de Computadores

**Sintonização de controladores difusos
através de técnicas de optimização não-linear
com restrições**

Dissertação para obtenção do Grau de
Mestre em Engenharia Electrotécnica e de Computadores

Orientador: Paulo José Carrilho de Sousa Gil, Professor Auxiliar,
Universidade Nova de Lisboa

Júri:

Presidente:	Luís Filipe Santos Gomes
Arguente:	José António Barata de Oliveira
Vogal:	Paulo José Carrilho de Sousa Gil



FACULDADE DE
CIÊNCIAS E TECNOLOGIA
UNIVERSIDADE NOVA DE LISBOA

Março, 2015

Sintonização de controladores difusos através de técnicas de optimização não-linear com restrições

Copyright © Ana Margarida Varela Sebastião, Faculdade de Ciências e Tecnologia, Universidade Nova de Lisboa

A Faculdade de Ciências e Tecnologia e a Universidade Nova de Lisboa têm o direito, perpétuo e sem limites geográficos, de arquivar e publicar esta dissertação através de exemplares impressos reproduzidos em papel ou de forma digital, ou por qualquer outro meio conhecido ou que venha a ser inventado, e de a divulgar através de repositórios científicos e de admitir a sua cópia e distribuição com objectivos educacionais ou de investigação, não comerciais, desde que seja dado crédito ao autor e editor.

Aos meus pais, irmão e avós

AGRADECIMENTOS

Em primeiro lugar gostaria de agradecer ao meu orientador, o Professor Doutor Paulo Gil pela orientação e disponibilidade que demonstrou ao longo do desenvolvimento desta dissertação.

Gostaria de agradecer também ao Professor Doutor Luís Palma pelo apoio dado no âmbito desta dissertação.

Agradecer em geral à Faculdade de Ciências e Tecnologia da Universidade Nova de Lisboa, e principalmente ao departamento de Engenharia Eletrotécnica por todo o conhecimento transmitido.

Ao meu colega e amigo Hugo Martins quero deixar o meu apreço, pois mostrou-se sempre disponível para o esclarecimento de dúvidas quando estas surgiam, contribuindo assim para a finalização da dissertação.

Também gostava de deixar a minha gratidão aos meus amigos que me acompanharam ao longo deste percurso académico e pessoal, onde partilhámos momentos fantásticos e únicos que irei recordar para sempre. Com a vossa presença os meus dias tornam-se sempre repletos de alegria e boa disposição, João de Abreu, Hugo Martins, Amanda Cabrera, Denise Sardinha e Dilcarina Sebastião.

Um agradecimento especial ao Emanuel Martins pelo amor, força, atenção e apoio dados ao longo deste percurso académico.

Por último gostava de deixar um agradecimento muito especial para o mais importante na minha vida, os meus pais, irmão e avós, que me apoiaram em todas as ocasiões e que tornaram este momento possível.

RESUMO

A presente dissertação tem como objetivo principal a implementação de uma arquitetura baseada em algoritmos evolutivos para a sintonização dos parâmetros do controlador PID (Proporcional-Integral-Derivativo) difuso, sendo o conceito de desempenho em malha fechada explicitamente tido em conta.

A sintonização dos parâmetros do controlador difuso é realizada tendo em conta um problema de otimização com restrições, em que a função de custo a ser minimizada é descrita em termos do desempenho em malha fechada, com a dinâmica do sistema a ser aproximada por um modelo não linear.

Como nas metodologias de otimização existentes, a incorporação de mecanismos de adaptação referentes às funções de pertinência não é comum, na presente dissertação é tido em conta, para além da usual sintonização dos fatores de escala, a sintonização dos fatores de escala e funções de pertinência em simultâneo.

Os resultados experimentais realizados num sistema de referência, visam demonstrar os benefícios de incorporar as funções de pertinência no processo de otimização em diferido. É também utilizado um método analítico de segunda ordem como referência, por forma a comparar o desempenho de uma abordagem de otimização global contra uma de otimização local. Finalmente é implementada uma abordagem em-linha, usando o método analítico de segunda ordem, na otimização dos fatores de escala e funções de pertinência.

Palavras-chave: Otimização global; Algoritmos evolutivos; Algoritmo evolutivo diferencial; Controlo lógico difuso; PID difuso.

ABSTRACT

The main goal of this thesis is the implementation of an architecture based on a evolutionary algorithms for tuning the parameters of a fuzzy PID (Proportional-Integral-Derivative), where the closed loop performance is explicitly taken into account.

The tuning of the fuzzy controller parameters is performed considering a constrained optimisation problem, where the cost function to be minimised is described in terms of the closed loop response, with the system dynamics described by a nonlinear model.

Since in existing optimisation methodologies, the incorporation of adaptation mechanisms regarding the membership functions is not common, in this thesis, besides the usual scaling factors tuning, the tuning of both scaling factors and membership functions is also taken into account.

Experimental results carried out on a benchmark system, aim to provide a clear picture of the benefits of favouring the optimisation, carried out in an offline configuration, for both scaling factors and membership functions. A second order analytical method is used in order to compare the performance of using a global optimisation approach against a local one. Finally the optimisation of both scaling factors and membership functions is implemented by considering an online approach, using the second order analytical method.

Keywords: Global optimisation; Evolutionary algorithms; Differential Evolution algorithm; Fuzzy logic control; Fuzzy PID.

CONTEÚDO

Conteúdo	xiii
Lista de Figuras	xv
Lista de Tabelas	xvii
1 Introdução	1
1.1 Motivação	1
1.2 Objetivos e contribuições	2
1.3 Organização da dissertação	2
2 Sistemas difusos	5
2.1 Evolução histórica	5
2.2 Lógica difusa	6
2.2.1 Conjuntos difusos	6
2.3 Raciocínio aproximado	11
2.3.1 Variáveis linguísticas	11
2.3.2 Proposições difusas	13
2.3.3 Proposições difusas <i>Se-Então</i>	13
2.4 Estrutura de um controlador difuso	14
2.4.1 Interface de fuzificação	14
2.4.2 Base de conhecimento	15
2.4.3 Mecanismos de inferência	17
2.4.4 Interface de desfuzificação	20
3 Controladores difusos do tipo Mamdani	21
3.1 Controladores PID difusos	21
3.1.1 Controlador PD difuso	22
3.1.2 Controlador PI difuso	24
3.1.3 Controlador PID difuso baseado em PI+PD	25
4 Modelação experimental de sistemas	27
4.1 Modelação linear	28
4.1.1 Estimação em diferido dos parâmetros do modelo ARX	29

4.1.2	Estimação recursiva dos parâmetros do modelo ARX	30
4.2	Modelação não linear	32
4.2.1	Estimação em diferido dos parâmetros da rede neuronal	33
4.2.2	Estimação em-linha dos parâmetros do modelo NNARX	35
4.3	Validação de modelos	36
4.3.1	Validação cruzada	36
4.3.2	Teste dos resíduos	37
5	Técnicas de otimização	39
5.1	Ótimo local e ótimo global	40
5.2	Métodos de otimização	41
5.2.1	Otimização sem restrições	41
5.2.2	Otimização com restrições	43
6	Casos de estudo	47
6.1	Descrição do sistema	47
6.2	Projeto do controlador PID difuso	48
6.3	Otimização em diferido do controlador PID	50
6.3.1	PID difuso com sintonização dos fatores de escala	50
6.3.2	PID difuso com sintonização dos fatores de escala e funções de pertença em diferido	54
6.4	Otimização em-linha do controlador PID difuso usando o método analítico de segunda ordem	59
6.4.1	PID difuso com sintonização dos fatores de escala e funções de pertença em-linha	60
6.5	Análise de resultados	69
7	Conclusões e perspectivas futuras	71
7.1	Conclusões gerais	71
7.2	Perspectivas futuras	72
	Bibliografia	73

LISTA DE FIGURAS

2.1	Conjuntos que representam o IMC de uma pessoa adulta (a) Conjunto Crespo (b) Conjunto difuso.	7
2.2	Propriedades de um conjunto difuso.	9
2.3	Operações sobre conjuntos clássicos utilizando diagramas de Venn.	10
2.4	Funções de pertença associadas a uma variável linguística.	12
2.5	Configuração básica do controlador lógico difuso.	14
2.6	Grau de sobreposição e taxa de sobreposição.	16
2.7	Propriedades de um conjunto difuso.	19
3.1	Estrutura do controlador PD difuso.	23
3.2	Estrutura do controlador PI difuso.	24
3.3	Estrutura do controlador PID difuso.	26
3.4	Estrutura do controlador PID difuso simplificada.	26
4.1	Estrutura geral dos modelos lineares.	28
4.2	Estrutura de um neurónio.	32
4.3	Estrutura de uma rede neuronal NNARX.	33
4.4	Rede neuronal composta por M+1 camadas.	33
5.1	Exemplo de uma função com vários extremos.	40
5.2	Configuração básica do Algoritmo Evolutivo Diferencial.	45
6.1	Sistema de tanques AMIRA [®] DTS200.	48
6.2	<i>National Instruments USB-6009</i>	48
6.3	Funções de pertença de \tilde{e} , $\Delta\tilde{e}$ e \tilde{u}	49
6.4	Funções de pertença de $\Delta\tilde{u}$	49
6.5	Tanque 1 do sistema MIMO controlado por um controlador PID difuso com sintonização dos fatores de escala em diferido, considerando o AED.	52
6.6	Tanque 2 do sistema MIMO controlado por um controlador PID difuso com sintonização dos fatores de escala em diferido, considerando o AED.	52
6.7	Tanque 1 do sistema MIMO controlado por um controlador PID difuso com sintonização dos fatores de escala em diferido, considerando o método analítico de segunda ordem.	53

6.8	Tanque 2 do sistema MIMO controlado por um controlador PID difuso com sintonização dos fatores de escala em diferido, considerando o método analítico de segunda ordem.	53
6.9	Tanque 1 do sistema MIMO controlado por um controlador PID difuso com sintonização dos fatores de escala e das funções de pertinência em diferido, considerando o AED.	55
6.10	Tanque 2 do sistema MIMO controlado por um controlador PID difuso com sintonização dos fatores de escala e das funções de pertinência em diferido, considerando o AED.	56
6.11	Funções de pertinência de \tilde{e} , $\Delta\tilde{e}$ e \tilde{u}	56
6.12	Funções de pertinência de $\Delta\tilde{u}$	57
6.13	Tanque 1 do sistema MIMO controlado por um controlador PID difuso com sintonização dos fatores de escala e das funções de pertinência em diferido, considerando o método analítico de segunda ordem.	57
6.14	Tanque 2 do sistema MIMO controlado por um controlador PID difuso com sintonização dos fatores de escala e das funções de pertinência em diferido, considerando o método analítico de segunda ordem.	58
6.15	Funções de pertinência de \tilde{e} , $\Delta\tilde{e}$ e \tilde{u}	59
6.16	Funções de pertinência de $\Delta\tilde{u}$	59
6.17	Tanque 1 do sistema MIMO controlado por um controlador PID difuso com sintonização dos fatores de escala e das funções de pertinência em-linha.	63
6.18	Tanque 2 do sistema MIMO controlado por um controlador PID difuso com sintonização dos fatores de escala e das funções de pertinência em-linha.	63
6.19	Evolução dos fatores de escala associados ao Tanque 1.	64
6.20	Evolução dos fatores de escala associados ao Tanque 2.	64
6.21	Evolução das larguras das funções de pertinência associadas ao Tanque 1.	65
6.22	Evolução das larguras das funções de pertinência associadas ao Tanque 2.	65
6.23	Tanque 1 do sistema MIMO controlado por um controlador PID difuso com sintonização dos fatores de escala e das funções de pertinência em-linha, considerando uma configuração em cascata.	66
6.24	Tanque 2 do sistema MIMO controlado por um controlador PID difuso com sintonização dos fatores de escala e das funções de pertinência em-linha, considerando uma configuração em cascata.	66
6.25	Evolução dos fatores de escala associados ao Tanque 1, considerando uma configuração em cascata.	67
6.26	Evolução dos fatores de escala associados ao Tanque 2, considerando uma configuração em cascata.	67
6.27	Evolução das larguras das funções de pertinência associadas ao Tanque 1, considerando uma configuração em cascata.	68
6.28	Evolução das larguras das funções de pertinência associadas ao Tanque 2, considerando uma configuração em cascata.	68

LISTA DE TABELAS

2.1	Funções de pertença mais comuns.	8
2.2	Implicações difusas.	14
2.3	Casos particulares da sobreposição de funções de pertença.	17
3.1	Base de regras genérica para o controlador PD difuso.	23
3.2	Base de regras genérica para o controlador PI difuso.	25
4.1	Algumas das estruturas mais utilizadas.	29
6.1	Larguras das funções de pertença.	49
6.2	Fatores de escala ótimos com sintonização em diferido usando o AED.	51
6.3	Fatores de escala ótimos com sintonização em diferido usando o método analítico de segunda ordem.	54
6.4	Fatores de escala ótimos com sintonização em diferido dos fatores de escala e funções de pertença usando o AED.	55
6.5	Larguras ótimas das funções de pertença com sintonização em diferido usando o AED.	56
6.6	Fatores de escala ótimos com sintonização em diferido dos fatores de escala e funções de pertença usando o método analítico de segunda ordem.	58
6.7	Larguras ótimas das funções de pertença com sintonização em diferido usando o método analítico de segunda ordem.	58
6.8	Métricas de desempenho.	69

LISTA DE ACRÓNIMOS

AE Algoritmos Evolutivos.

AED Algoritmo Evolutivo Diferencial.

ARMA Auto Regressive Moving Average.

ARMAX Auto Regressive Moving Average with eXogenous input.

ARX Auto Regressive with eXogenous input.

BJ Box Jenkins.

CLD Controllo Lógico Difuso.

filter-SQP filter-Sequential Quadratic Programming.

IMC Índice de Massa Corporal.

KKT Karush-Kuhn-Tucker.

LSSQP Large Scale Sequential Quadratic Programming.

MIMO Multi-Input Multi-Output.

NARMA Nonlinear Auto Regressive Moving Average.

NARX Nonlinear Auto Regressive with eXogenous input.

NNARX Neural Network Auto Regressive with eXogenous input.

NOE Nonlinear Output Error.

OE Output Error.

PD Proporcional-Derivativo.

PI Proporcional-Integral.

PID Proporcional-Integral-Derivativo.

PQ Programação Quadrática.

PQS Programação Quadrática Sequencial.

RMSE Root Mean Square of Error.

RMSI Root Mean Square of control action Increment.

SNOPT Sparse Nonlinear Optimizer.

TCP/IP Transmission Control Protocol/Internet Protocol.

INTRODUÇÃO

Na aplicação de sistemas de controlo, o controlo Proporcional-Integral-Derivativo (PID) é ainda nos dias de hoje umas das técnicas mais populares encontradas na indústria. A motivação principal para o uso desta metodologia está relacionada com o facto de ser simples, tanto a nível funcional como estrutural, ser fácil de sintonizar e ter um baixo custo de implementação. Contudo, quando a dinâmica do sistema é difícil de caracterizar ou se encontra sujeita a incertezas, pode-se tornar complicado usar as técnicas de controlo convencional. Por este motivo, nos últimos anos, o controlo de sistemas que apresentam alguma complexidade, designadamente incertezas ou não linearidades, tornou-se um tópico de importância considerável na literatura [Kum+11].

1.1 Motivação

Quando os sistemas de controlo PID convencionais lidam com sistemas lineares ou até mesmo com dinâmicas ligeiramente não lineares, conseguem fornecer teoricamente um desempenho aceitável em anel fechado. No entanto, o mesmo não se verifica para o caso em que as não linearidades são moderadas ou elevadas, resultando num mau desempenho em malha fechada ou até mesmo na instabilidade do sistema em anel fechado.

Nas situações em que o sistema é não linear, ainda existe uma forte motivação para escolher topologias de controlo topologicamente similares às do tipo PID. A teoria da lógica difusa fornece neste contexto as ferramentas necessárias para a conceção de tais estruturas de controlo, em particular os controladores PID difusos do tipo Mamdani [Fen06]. Este tipo de controladores é considerado uma boa metodologia, pois origina melhores resultados face ao controlo convencional.

As técnicas de controlo lógico difuso (CLD) representam a aplicação da perícia e conhecimento humano para lidar eficazmente com sistemas complexos e não lineares. Estas proporcionam um meio eficaz de capturar a natureza aproximada e inexata do mundo real. Por este motivo, a parte essencial do CLD traduz-se num conjunto de estratégias de controlo linguístico baseadas em conhecimento qualitativo sobre o comportamento do sistema [Kum+11].

Na literatura é possível encontrar várias abordagens para a sintonização de controladores difusos, incluindo metodologias baseadas em heurísticas [Mis+96], métodos dependentes da pseudo-equivalência entre controladores PID difusos e convencionais [Bou+10] ou métodos que se baseiam em técnicas de otimização, como por exemplo algoritmos evolutivos [HL09].

1.2 Objetivos e contribuições

O objetivo principal deste trabalho consiste na implementação de uma metodologia baseada em técnicas de computação evolutiva, a qual recorre a algoritmos evolutivos, para a sintonização dos parâmetros do controlador PID difuso.

Será tido em conta, para além da habitual sintonização em diferido dos fatores de escala do controlador PID difuso, a sintonização em diferido e em simultâneo dos fatores de escala e funções de pertença, pois esta prática é pouco comum na literatura.

Por forma a demonstrar as vantagens intrínsecas de uma abordagem de otimização global face a uma de otimização local, o método evolutivo estudado nesta dissertação será comparado com um método analítico de segunda ordem.

Por fim, a incorporação da adaptação das funções de pertença no processo de otimização é também realizada através de uma implementação em linha, usando um algoritmo analítico de segunda ordem.

1.3 Organização da dissertação

Esta dissertação encontra-se organizada em sete capítulos, incluindo o presente.

No capítulo 2 são introduzidos os conceitos básicos da lógica difusa e da teoria dos conjuntos difusos, realçando as características que os distinguem dos conjuntos clássicos. O conceito de raciocínio aproximado é também abordado neste capítulo, introduzindo os conceitos de variável linguística e proposições difusas. Finalmente é apresentada a estrutura geral de um controlador difuso, juntamente com a descrição de cada um dos

seus componentes.

No capítulo 3 é descrito o controlador PID difuso do tipo Mamdani, que corresponde a uma agregação de dois outros controladores, nomeadamente o PI e o PD difusos.

O capítulo 4 aborda a modelação experimental de sistemas lineares e não lineares, exemplificando alguns métodos de estimação de parâmetros, tanto em diferido como em linha, apresentando-se ainda os métodos de validação de modelos mais comuns.

No capítulo 5 são abordados os conceitos de ótimo local e ótimo global e discutidos alguns métodos de otimização com e sem restrições.

O capítulo 6 apresenta os resultados experimentais realizados num sistema de referência, juntamente com a sua descrição. É realizado um estudo comparativo do desempenho dos controladores PID difusos sintonizados através de seis abordagens diferentes, em que as primeiras quatro são implementadas em diferido e as últimas duas metodologias com uma configuração em linha. Inclui-se ainda uma análise dos resultados obtidos.

No capítulo 7 são apresentadas as principais conclusões decorrentes dos trabalhos efetuados no âmbito desta dissertação, sugerindo-se ainda algumas perspetivas futuras que visam complementar o trabalho desenvolvido.

SISTEMAS DIFUSOS

A lógica difusa pode ser considerada como a lógica precisa da imprecisão e do raciocínio aproximado [Zad08]. Uma das principais vantagens em usar esta álgebra no projeto de controladores está relacionado com a sua facilidade de projeto, que é em certa medida semelhante à forma de pensamento humano [Nov95], possibilitando, facilmente, a incorporação de conhecimento de especialistas [AOAL08]. Por outro lado, a lógica difusa torna fácil a descrição e análise de sistemas que possam ser muito complexos ou mal definidos para admitirem uma análise matemática precisa. O Controlo Lógico Difuso (CLD) consiste num conjunto de regras difusas relacionadas com o conceito duplo de implicações difusas e regras composicionais de inferência. Este esquema foi introduzido por Lofti A. Zadeh¹ e é conhecido pela sua habilidade em lidar com não linearidades e incertezas [Zad65]. Um sistema difuso contém um conjunto de regras ou associações sob a forma "Se as condições antecedentes se mantêm, então as condições consequentes mantêm-se" [Kos92].

2.1 Evolução histórica

A lógica difusa surgiu em 1965 com a introdução do conceito de conjuntos difusos por Lofti Zadeh, sendo este capaz de tratar o aspecto vago da informação. A primeira aplicação no domínio do controlo difuso remonta a 1974, quando Ebrahim Mamdani aplicou o conceito de controlo lógico difuso para controlar um motor a vapor [Riz+11]. Desde então, é possível encontrar uma grande variedade de aplicações com este tipo de álgebra, nomeadamente na modelação qualitativa, reconhecimento de padrões, processamento de sinais, processamento de informação, finanças, gestão, medicina e robótica, entre outros [Fen06].

¹Professor de Ciências da Computação na Universidade de Berkeley na Califórnia

2.2 Lógica difusa

A lógica difusa aproxima-se bastante do pensamento humano e da linguagem natural. Basicamente, esta lógica fornece um meio eficaz para capturar a aproximada, inexata natureza do mundo real [Lee90a].

2.2.1 Conjuntos difusos

A teoria dos conjuntos difusos tem por base a teoria clássica dos conjuntos. Pode-se interpretar a álgebra de Boole, desenvolvida por George Boole ², em termos da teoria clássica dos conjuntos, que atribui apenas dois estados de verdade (0-falso ou 1-verdadeiro) e que passou a ser conhecida como Lógica de Boole [Dha11].

Na teoria de conjuntos difusos de Zadeh [Zad65], um conjunto difuso corresponde a uma coleção de elementos que têm características comuns. Na lógica de boole é possível dizer que um objecto pertence ou não pertence a um conjunto, e dado este facto o conjunto possui um grau de pertença que assume 0 ou 1 [Riz+11]. No domínio da teoria dos conjuntos difuso, os conjuntos booleanos são intitulados por conjuntos "crespos". Para um conjunto "crespo" C no universo de discurso X [Asc95; Ros09] é possível definir a seguinte função característica $\mu_C : X \rightarrow \{0;1\}$, isto é:

$$\mu_C(x) = \begin{cases} 1 & \text{se } x \in C \\ 0 & \text{se } x \notin C \end{cases} \quad (2.1)$$

Exemplo 2.1

Na figura 2.1 encontram-se representadas as funções características do caso tomado como exemplo, adultos com Índice de Massa Corporal (IMC) normal. Como se pode verificar, de acordo com a lógica booleana, figura 2.1 (a), um indivíduo adulto tem um IMC normal caso este se encontre no intervalo $[18,5; 25]$. Caso o IMC ultrapasse um pouco este intervalo (por exemplo umas décimas) o indivíduo passa automaticamente a não ter um IMC considerado normal. A função característica que representa o conjunto difuso, figura 2.1 (b), demonstra que a transição abrupta na figura 2.1 (a) é substituída por uma transição mais suave [Riz+11].

As funções de pertença são curvas que definem a forma como cada ponto no espaço de entrada é mapeado para um valor de pertença (ou grau de pertença) entre $[0;1]$. O conjunto definido com base nesta função de pertença é designado por conjunto difuso [PP10].

²(2 de Novembro de 1815 - 8 de Dezembro de 1864), foi um matemático e filósofo britânico responsável pela criação da Álgebra Booleana, fundamental para o desenvolvimento da computação moderna

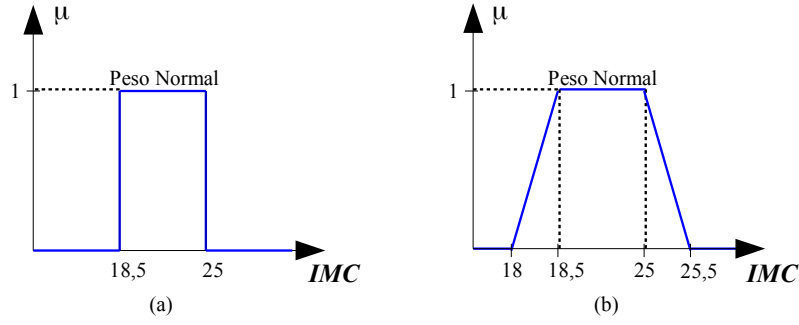


Figura 2.1: Conjuntos que representam o IMC de uma pessoa adulta (a) Conjunto Crespo (b) Conjunto difuso.

Um conjunto difuso F no universo de discurso U é caracterizado por uma função de pertença μ_F , a qual toma valores no intervalo $[0; 1]$,

$$\mu_F : U \rightarrow [0; 1] \quad (2.2)$$

Um conjunto difuso pode ser visto como uma generalização do conceito de um conjunto ordinário, cujas funções de pertença apenas tomam dois valores (0;1). Assim um conjunto difuso F em U pode ser representado por um conjunto de pares ordenados de um elemento genérico u e pelo seu grau da função de pertença μ_F [Lee90b]:

$$F = \{(u, \mu_F(u)) \mid u \in U\} \quad (2.3)$$

Quando U é discreto, o conjunto difuso F é representado por:

$$F = \sum_{i=1}^n \mu_F(u_i) / u_i \quad (2.4)$$

No caso de U ser um universo de discurso contínuo, F é representado por:

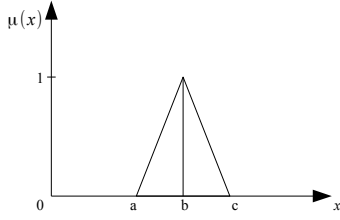
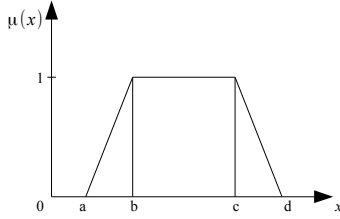
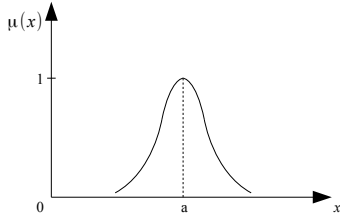
$$F = \int_U \mu_F(u) / u \quad (2.5)$$

Existem inúmeros tipos de funções de pertença, tais como do tipo triangular, trapezoidal, curvas em forma de S, curvas em forma de Π , Gaussiana e funções do tipo sigmoide. De todas estas funções, as mais comuns são as do tipo triangular, trapezoidal e gaussiana. A tabela 2.1 apresenta estas funções de pertença, assim como as respectivas equações características [Lee06].

2.2.1.1 Propriedades dos conjuntos difusos

O conceito de convexidade é uma propriedade importante dos conjuntos difusos, sendo particularmente útil em aplicações que envolvam classificação de padrões, otimização e

Tabela 2.1: Funções de pertença mais comuns.

Função de pertença	Caracterização
<p>Triangular</p> 	$\Lambda : X \rightarrow [0; 1]$ $\Lambda(x, a, b, c) = \begin{cases} 0 & x < a \\ \frac{x-a}{b-a} & a \leq x \leq b \\ \frac{c-x}{c-b} & b \leq x \leq c \\ 0 & x > c \end{cases}$
<p>Trapezoidal</p> 	$\Pi : X \rightarrow [0; 1]$ $\Pi(x, a, b, c, d) = \begin{cases} 0 & x < a \\ \frac{x-a}{b-a} & a \leq x \leq b \\ 1 & b \leq x \leq c \\ \frac{d-x}{d-c} & c \leq x \leq d \\ 0 & x > d \end{cases}$
<p>Gaussiana</p> 	$\Omega : X \rightarrow [0; 1]$ $\Omega(x, a, \sigma) = \exp\left(-\frac{(x-a)^2}{2\sigma^2}\right)$

outros problemas relacionados. Um dado conjunto difuso A é convexo se a sua função de pertença for crescente, decrescente ou em forma de sino, ou seja [Dri+93; Zad65]:

$$\forall x_1, x_2 \in X, \forall \lambda \in [0; 1] : \mu_A(\lambda x_1 + (1 - \lambda)x_2) \geq \min(\mu_A(x_1), \mu_A(x_2)) \quad (2.6)$$

Os conjuntos difusos apresentam três propriedades principais: suporte, altura e núcleo. A figura 2.2 apresenta um conjunto difuso com a identificação das propriedades mencionadas, com base na referência [Dri+93].

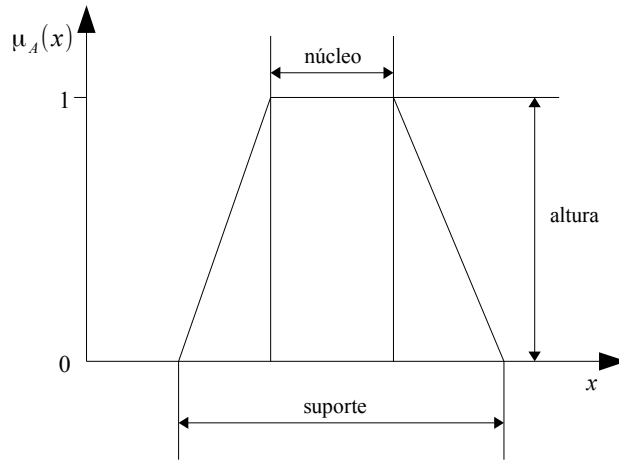


Figura 2.2: Propriedades de um conjunto difuso.

Suporte

O suporte de um conjunto difuso A corresponde a um conjunto *crespo* que contém todos os elementos $x \in X$ com grau de pertinência não nulo, ou seja:

$$\text{suporte}(A) = \{x \in X : \mu_A(x) > 0\} \quad (2.7)$$

Sendo o conjunto A um conjunto que satisfaz a propriedade de convexidade, pode-se dizer que o seu suporte é um intervalo, também designado por *largura*. A *largura* de um conjunto difuso convexo é definida por:

$$\text{largura}(A) = \sup(\text{suporte}(A)) - \inf(\text{suporte}(A)) \quad (2.8)$$

onde *sup* e *inf* referem-se às operações matemáticas *supremo* e *infimo*, respetivamente. Caso o conjunto $\text{suporte}(A)$ seja limitado, como é usual em controlo difuso, *sup* e *inf* podem ser substituídos por *max* (máximo) e *min* (mínimo).

Altura

A altura de um conjunto difuso A é igual ao maior grau de pertinência e é definida por:

$$\text{altura}(A) = \sup_{x \in X} \mu_A(x) \quad (2.9)$$

Núcleo

O núcleo de um conjunto difuso A é definido como a região do universo cujo grau de pertinência em A é 1:

$$\text{núcleo}(A) = \{x \in X : \mu_A(x) = 1\} \quad (2.10)$$

2.2.1.2 Operações sobre conjuntos difusos

O diagrama de Venn é utilizado para representar todas as relações matemáticas ou lógicas, tais como a interseção, união e complemento, entre diferentes grupos de conjuntos. Na lógica clássica é comum usar-se este tipo de diagramas para representar o conjunto universo X e os conjuntos formados a partir dos seus elementos. Na figura 2.3 a zona sombreada a azul representa as operações efetuadas sobre os conjuntos A e B no universo X .

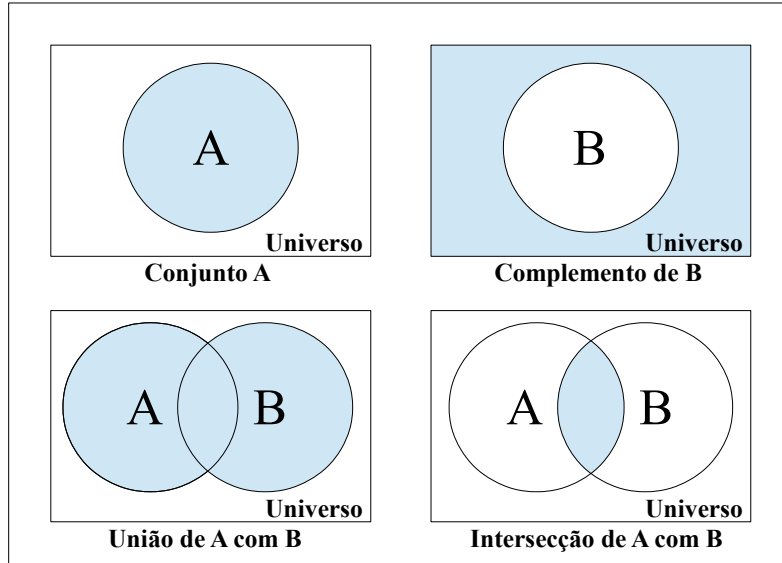


Figura 2.3: Operações sobre conjuntos clássicos utilizando diagramas de Venn.

A igualdade de dois conjuntos difusos ou a inclusão de um conjunto dentro de outro conjunto difuso (subconjunto) são algumas das noções da teoria clássica de conjuntos que podem ser aplicadas aos conjuntos difusos. Sejam A e B conjuntos difusos definidos no universo X :

O conjunto A é considerado igual ao conjunto B ($A = B$) sse:

$$\forall x \in X : \mu_A(x) = \mu_B(x) \quad (2.11)$$

O conjunto A é um subconjunto de B ($A \subseteq B$) sse:

$$\forall x \in X : \mu_A(x) \leq \mu_B(x) \quad (2.12)$$

No caso dos conjuntos difusos, existem diferentes formas de definir as operações de união, interseção e complemento, através do uso dos operadores de Zadeh ou Yager, entre outros. Nesta secção serão apenas abordados os operadores de Zadeh.

Operadores de Zadeh

Os operadores propostos por Zadeh para as operações de união, intersecção e complemento de conjuntos difusos são os seguintes [Zad73]:

Intersecção de conjuntos

$$\forall x \in X : \mu_{A \cap B}(x) = \min(\mu_A(x), \mu_B(x)) \quad (2.13)$$

União de conjuntos

$$\forall x \in X : \mu_{A \cup B}(x) = \max(\mu_A(x), \mu_B(x)) \quad (2.14)$$

Complemento de um conjunto

$$\forall x \in X : \mu_{\bar{A}}(x) = 1 - \mu_A(x) \quad (2.15)$$

2.3 Raciocínio aproximado

O raciocínio aproximado representa um modo de raciocínio que nem é exato nem muito inexato, o qual pode oferecer uma estrutura de raciocínio humano mais realista do que a tradicional lógica binária [Zad75]. Este tipo de raciocínio é o mais conhecido na lógica difusa e abrange uma variedade de regras de inferência cujas regras contêm proposições difusas. A consequência de um dado conjunto de proposições depende fundamentalmente do significado destas proposições [Dri+93].

2.3.1 Variáveis linguísticas

As variáveis linguísticas são variáveis em que os valores não são números mas sim palavras em linguagem natural. A motivação que leva ao uso de palavras em vez de números é que a caracterização linguística é geralmente menos específica do que a numérica [Zad75].

Exemplo 2.2

Considerando como exemplo a propriedade "*idade*", quando se diz "*o Hugo é novo*" é-se menos preciso do que quando se diz, "*o Hugo tem 25 anos*". Neste contexto, a classificação "*novo*" pode ser considerada como um *valor linguístico* da variável "*idade*". Ambas as classificações têm o mesmo objetivo no entanto pode-se inferir que o valor linguístico é menos preciso e, portanto, menos informativo.

Estas variáveis linguísticas têm como principal objetivo fornecer um meio que permita descrever sistemas que sejam demasiado complexos ou mal definidos para admitir uma análise matemática precisa [Zad73].

Uma variável linguística é definida por um quadripleto $(X, L(X), X, M)$ em que [Zad75]:

- X corresponde à variável linguística;
- LX é o conjunto dos valores (termos) linguísticos que a variável X pode tomar;
- X é o universo de discurso;
- M é a regra semântica que associa a cada valor linguístico X o seu significado $M(X)$, num subconjunto de X .

Considere-se no caso de um controlador difuso a variável linguística erro. Esta variável é definida pelo quadripleto $(E, LE, E, M(E))$, onde,

- E , representa a variável linguística erro;
- $LE, \{NG, NM, NP, ZO, PP, PM, PG\}$ o conjunto de termos linguísticos ou conjuntos de referência da variável linguística erro;
- $E, [-1,5; 1,5]$ o universo de discurso;
- $M(E)$, significado de cada termo linguístico da variável linguística erro, através da função de pertença associada a cada um dos termos³ (ver Figura 2.4).

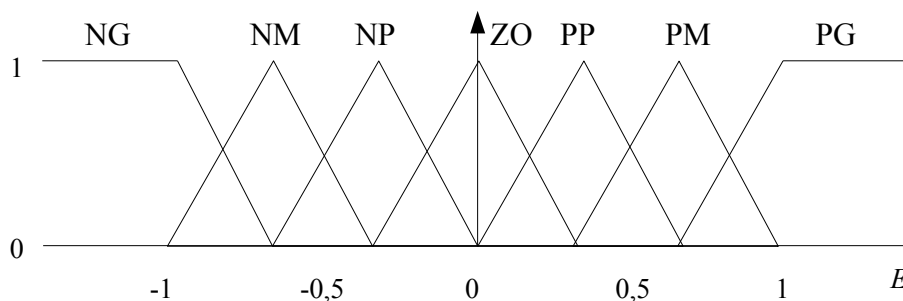


Figura 2.4: Funções de pertinência associadas a uma variável linguística.

³NG = Negativo Grande, NM = Negativo Médio, NP = Negativo Pequeno, ZO = Zero, PP = Positivo Pequeno, PM = Positivo Médio; PG = Positivo Grande

2.3.2 Proposições difusas

Existem dois tipos de proposições difusas, designadamente, proposições difusas atómicas e proposições difusas compostas [Wan99].

Uma proposição difusa atómica representa uma declaração única, como por exemplo:

$$x \text{ é } A \quad (2.16)$$

em que x é a variável linguística, e A é o valor linguístico de x , ou seja, A é um conjunto difuso definido no domínio físico de x .

As proposições difusas são proposições mais complexas, construídas com base na noção de proposições difusas atómicas e nas conectividades linguísticas *e*, *ou*, *não* (as quais representam a intersecção, união e complemento, respetivamente) e *se – então*.

Exemplo 2.3

Tomando como exemplo a variável linguística erro, representada pelo símbolo E , as seguintes expressões são exemplos de proposições difusas, sendo as primeiras três referentes a proposições difusas atómicas e as últimas três a proposições difusas compostas.

$$E \text{ é } NG \quad (2.17)$$

$$E \text{ é } NM \quad (2.18)$$

$$E \text{ é } NP \quad (2.19)$$

$$E \text{ é } NG \text{ ou } E \text{ não é } NM \quad (2.20)$$

$$E \text{ não é } NG \text{ e } E \text{ não é } NP \quad (2.21)$$

$$(E \text{ é } NG \text{ e } E \text{ não é } NP) \text{ ou } E \text{ é } NM \quad (2.22)$$

onde NG , NM e NP significam "*Negativo Grande*", "*Negativo Médio*" e "*Negativo Pequeno*", respetivamente.

2.3.3 Proposições difusas Se-Então

As proposições difusas do tipo [Dri+93]

$$\text{Se } X \text{ é } A \text{ então } Y \text{ é } B \quad (2.23)$$

são interpretadas como relações difusas, as quais podem ter inúmeras interpretações sobre o seu significado. Nesta secção serão apresentadas algumas das implicações difusas mais utilizadas (ver Tabela 2.2), que traduzem algumas das possíveis interpretações da regra representada pela equação (2.23) [BJ08; Wan99].

Tabela 2.2: Implicações difusas.

Lukasiewicz	$I_{LK}(a, b) = (\min(1, 1 - a + b))$
Gödel	$I_{GD}(a, b) = \begin{cases} 1, & \text{se } a \leq b \\ b, & \text{se } a > b \end{cases}$
Reichenbach	$I_{RC}(a, b) = 1 - a + ab$
Kleene-Dienes	$I_{KD}(a, b) = \max(1 - a, b)$
Goguen	$I_{GG}(a, b) = \begin{cases} 1, & \text{se } a \leq b \\ \frac{b}{a}, & \text{se } a > b \end{cases}$
Rescher	$I_{RS}(a, b) = \begin{cases} 1, & \text{se } a \leq b \\ 0, & \text{se } a > b \end{cases}$
Yager	$I_{YG}(a, b) = \begin{cases} 1 & \text{se } a=0 \text{ e } b=0 \\ b^a & \text{se } a>0 \text{ ou } b>0 \end{cases}$
Weber	$I_{WB}(a, b) = \begin{cases} 1, & \text{se } a < 1 \\ b, & \text{se } a = 1 \end{cases}$
Fodor	$I_{FD}(a, b) = \begin{cases} 1, & \text{se } a \leq b \\ \max(1 - a, b), & \text{se } a > b \end{cases}$
Zadeh	$I_{ZD}(a, b) = \min(1 - a, \min(a, b))$
Mamdani	$I_{MD}(a, b) = \min(a, b)$

2.4 Estrutura de um controlador difuso

A estrutura básica de um sistema de CLD é constituída por quatro componentes conceptuais: a base de conhecimento, a interface de fuzificação, o mecanismo de inferência e, finalmente, a interface de desfuzificação [Fen06]. A configuração básica do controlador difuso encontra-se esquematizada na figura 2.5.

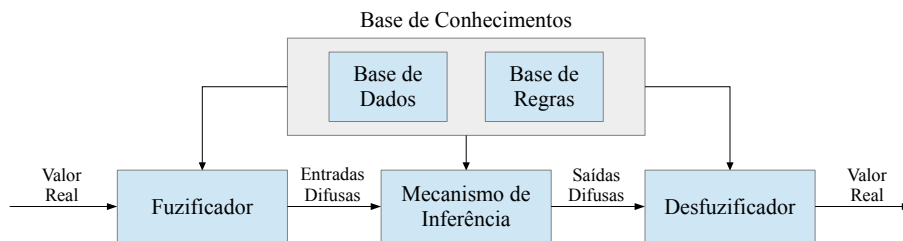


Figura 2.5: Configuração básica do controlador lógico difuso.

2.4.1 Interface de fuzificação

A componente de fuzificação define um mapeamento a partir de um subespaço de valores reais (crespos) para um difuso [Fen06], ou seja, um operador de fuzificação tem o poder de transformar dados crespos em conjuntos difusos, por forma a torná-los compatíveis com a representação interna do controlador. Em termos simbólicos, este operador pode

ser representado por:

$$x = \text{fuzificador}(x_0) \quad (2.24)$$

onde x_0 é o valor de entrada crespo do processo; x conjunto difuso; *fuzificador* representa um operador de fuzificação [Lee90b].

Os conjuntos difusos são interpretados como funções de pertença μ_X que associam a cada elemento x do universo de discurso U , um número $\mu_X(x)$ no intervalo $[0,1]$. Existem dois tipos principais de operadores de fuzificação, nomeadamente o *singleton* e o *nonsingleton*. No caso do fuzificador *singleton*, este gera, a partir de uma entrada crespa, um conjunto difuso *singleton* [Riz+11], ou seja:

$$\mu_X(x_i) = \begin{cases} 1 & \text{se } x_i = x \\ 0 & \text{se } x_i \neq x \end{cases} \quad (2.25)$$

O valor crespo x é convertido num conjunto difuso X com suporte x_i onde $\mu_X(x_i) = 1$, para $x_i = x$ e $\mu_X(x_i) = 0$ para $x_i \neq x$.

No caso do fuzificador *nonsingleton* o valor crespo x é convertido num conjunto difuso X com suporte x_i onde μ_X alcança o valor máximo em $x_i = x$ e diminui quando se afasta de $x_i = x$. Este fuzificador é útil em casos onde, por exemplo, os dados possam estar corrompidos por ruído [MM97].

2.4.2 Base de conhecimento

A base de conhecimento contém toda a informação necessária ao controlador e é constituído por dois módulos, nomeadamente a base de regras e a base de dados. A base de regras é essencialmente a parte processual do conhecimento, i.e. define a estratégia de controlo a ser implementada, enquanto que a base de dados compreende a parte declarativa do conhecimento [Fen06].

Base de regras

Um sistema difuso é caracterizado por um conjunto de declarações linguísticas baseadas no conhecimento especializado, onde este tipo de conhecimento é normalmente expresso sob a forma de regras *se-então*. Usualmente, estas apresentam-se da seguinte forma:

Se (um conjunto de condições são satisfeitas)

Então (um conjunto de consequências podem ser inferidas)

A declaração "*Se*()" de uma regra é designada por antecedente e apresenta-se como uma condição no seu domínio de aplicação. A parte subordinada ao "*Então*()" é denominada

consequente, a qual representa uma ação de controlo. O conjunto de regras de controlo difuso, que são expressas como declarações condicionais difusas, formam a base de regras do CLD. Basicamente estas regras proporcionam uma forma prática de expressar a política de controlo e o domínio de conhecimento. Refira-se, ainda, que estas regras são de fácil implementação [Lee90b].

Base de dados

A base de dados fornece as definições necessárias que são usadas para definir as regras de controlo linguístico, sendo composta pelas funções de pertença e pelos fatores de escala (ganhos do controlador) de um controlador difuso. É necessário que a escolha das funções de pertença seja realizada corretamente, pois estas desempenham um papel fulcral no que diz respeito ao bom funcionamento deste controlador.

Características das funções de pertença - Sobreposição e simetria

Quando duas funções de pertença se intersejam podemos dizer que estamos perante uma sobreposição, à qual está associada um grau de sobreposição. Este grau é designado por grau de pertença e corresponde ao ponto do universo de discurso em que se intersejam, sendo este um valor compreendido entre 0 e 1. O número de vezes que duas funções de pertença se intersejam é designado por taxa de sobreposição (ver Figura 2.6). A sobreposição é fundamental no controlo difuso, uma vez que confere robustez ao controlador [BM+96].

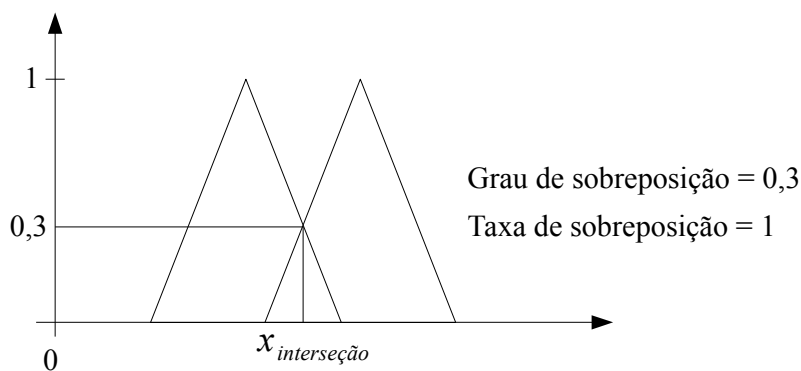


Figura 2.6: Grau de sobreposição e taxa de sobreposição.

Ainda em relação à sobreposição das funções de pertença, existem alguns casos particulares, tais como as situações em que o grau de sobreposição ou a taxa de sobreposição possam ser zero. Estas particularidades encontram-se discriminadas na tabela 2.3, e fazem-se acompanhar por uma descrição pormenorizada das implicações em termos de controlo de processos.

Tabela 2.3: Casos particulares da sobreposição de funções de pertença.

Caso particular	Implicação no sistema de controle
<p>Taxa de sobreposição zero</p>	<p>A existência de uma zona morta entre as funções de pertença cria descontinuidades na ação de controle, i.e., existem valores crespos de entrada para os quais nenhuma regra disparará, o que implica que nenhum valor será calculado para a saída do controlador, surgindo assim uma zona sem intervenção do controlador.</p>
<p>Grau de sobreposição zero</p>	<p>Dispara apenas uma regra de cada vez.</p>

De acordo com Kosko⁴ [Kos92], o grau de sobreposição entre funções de pertença deve estar compreendido entre 0,25 e 0,5, enquanto que a taxa de sobreposição entre funções de pertença deve ser exatamente 1, o que implica a sua simetria. Além disso, a escolha destes valores para o grau e taxa de sobreposição irá contribuir, em certa medida, para reduzir tanto o tempo de subida como o tempo de estabelecimento da resposta ao degrau.

2.4.3 Mecanismos de inferência

O mecanismo de inferência representa um modelo de raciocínio que realiza operações de inferência sobre as regras de controle difuso [Fen06], e a sua função principal traduz-se no cálculo do valor global da saída do controlador admitindo as contribuições individuais de cada regra da base de regras. Cada contribuição individual consiste no cálculo do valor da saída usando apenas uma única regra. Em relação à saída do módulo de fuzificação, a qual representa os valores difusos atuais do estado do processo, esta é comparada com cada uma das regras antecedentes, fazendo assim com que seja estabelecido um grau de correspondência para cada regra. Com base neste grau de correspondência obtém-se o valor difuso de saída [Dri+93].

⁴Bart Kosko (nascido a 7 de fevereiro de 1960) é um escritor e professor de engenharia elétrica na University of Southern California

Existem dois tipos de abordagens diferentes no que diz respeito ao projeto do mecanismo de inferência. O primeiro está relacionado com a inferência baseada na composição e o segundo com a inferência baseada em regras individuais. Em relação à primeira abordagem, todas as regras individuais pertencentes à base de regras são combinadas numa única relação difusa, a qual pode ser interpretada como uma única regra *se-então*. Em seguida o mecanismo de inferência realiza a operação de composição entre a entrada difusa e a relação difusa atrás mencionada. Desta forma obtém-se o conjunto difuso que descreve o valor difuso da saída. Relativamente à segunda abordagem, a inferência é aplicada individualmente a cada regra da base de regras, sendo os resultados de todas estas inferências agregados em seguida, por forma a criar o conjunto difuso que representa o valor difuso de saída. A segunda abordagem é a mais utilizada, pois é muito eficiente em termos computacionais e parcimonioso em termos de memória [Dri+93; Wan99].

Nesta secção considera-se o mecanismo de inferência do tipo Mamdani, pois além de ser uma das implicações difusas mais conhecidas na literatura, este foi aplicado no âmbito do desenvolvimento da presente dissertação.

2.4.3.1 Método de inferência do tipo Mamdani

A inferência do tipo Mamdani usa a implicação de Mamdani para representar o significado da regra *se-então*. Num sistema composto por n regras, cada regra é representada simbolicamente por [Dri+93]:

$$\text{Se } E \text{ é } LE^{(k)} \text{ então } U \text{ é } LU^{(k)}, k = 1, \dots, n$$

onde $LE^{(k)}$ e $LU^{(k)}$ representam os valores linguísticos das variáveis E e U , na regra k , respetivamente.

A interpretação de Mamdani sobre a regra é definida por:

$$\forall k : \tilde{R}_m^{(k)} = \int_{\text{ExU}} \min(\mu_{LE^{(k)}}(E), \mu_{LU^{(k)}}(U)) / (E, U) \quad (2.26)$$

A função de pertença da entrada cresspa E^* é dada por:

$$\forall E : \mu^*(E) = \begin{cases} 1 & \text{para } E = E^* \\ 0 & \text{outros} \end{cases} \quad (2.27)$$

O significado de todo o conjunto de regras é definido por:

$$\tilde{R}_m = \bigcup_{k=1}^n \tilde{R}_m^{(k)} \quad (2.28)$$

o que implica:

$$\forall E, U : \mu_{R_m}(E, U) = \max_k \mu_{R_m^{(k)}}(E, U) = \max_k \min(\mu_{LE^{(k)}}(E), \mu_{LU^{(k)}}(U)) \quad (2.29)$$

Como tal, a inferência de um conjunto de regras pode ser expressa por:

$$\tilde{U} = \mu^* \circ \tilde{R}_m \quad (2.30)$$

em que \circ representa a operação composição. No caso da inferência do tipo Mamdani é utilizada a composição *max-min*, fazendo com que a equação (2.30) tome a seguinte forma:

$$\forall U : \mu_U(U) = \max_k \min(\mu_{LE^{(k)}}(E^*), \mu_{LU^{(k)}}(U)) \quad (2.31)$$

Exemplo 2.4

Encontra-se apresentado na figura 2.7 um diagrama representativo do método de inferência de Mamdani [Lee90a; Lee90b] nas seguintes regras individuais:

$$R1 : \text{Se } E \text{ é } A_1 \text{ e } \Delta E \text{ é } B_1 \text{ então } \Delta U \text{ é } C_1 \quad (2.32)$$

$$R2 : \text{Se } E \text{ é } A_2 \text{ e } \Delta E \text{ é } B_2 \text{ então } \Delta U \text{ é } C_2 \quad (2.33)$$

em que E , ΔE , e ΔU representam as variáveis linguísticas erro, variação do erro e variação da ação de controlo, respetivamente; A_1 , B_1 , C_1 e A_2 , B_2 , C_2 são os conjuntos difusos associados aos termos linguísticos. Admite-se e_0 e Δe_0 como valores das variáveis de entrada do controlador difuso, por exemplo, o erro e a variação do erro no instante atual. Para os antecedentes de cada regra existe um grau de pertença, entre as entradas crespas e_0 e Δe_0 e os conjuntos difusos. Finalmente é utilizado o mecanismo de inferência do tipo Mamdani (*max-min*) para obter o valor global de saída do controlador difuso.

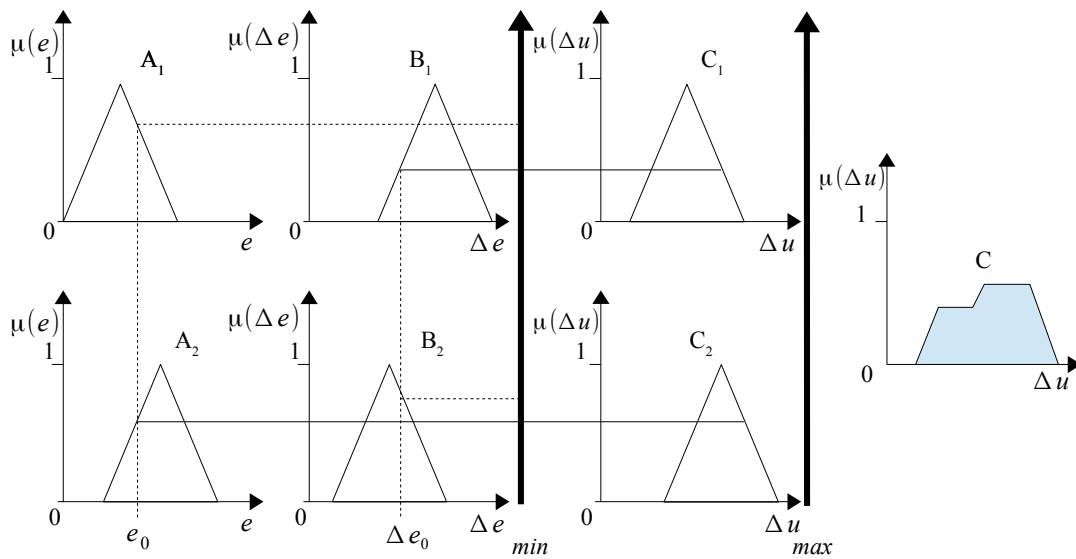


Figura 2.7: Propriedades de um conjunto difuso.

2.4.4 Interface de desfuzificação

A função do módulo de desfuzificação consiste em converter um conjunto difuso de saída num valor crespo [Lee90b], ou seja:

$$u_0 = \text{desfuzificador}(u) \quad (2.34)$$

onde u_0 representa o valor crespo e *desfuzificador* é o operador de desfuzificação.

Existem vários métodos de desfuzificação [DA11], sendo os mais comuns:

- **Centroide**

Esta técnica seleciona o valor crespo de saída correspondente ao centro de "massa" da função de pertença de saída. Esta é a técnica de desfuzificação mais utilizada e também a mais precisa. Esta técnica pode ser representada pela seguinte expressão:

$$u_0 = \frac{\int u \mu(u) du}{\int \mu(u) du} \quad (2.35)$$

onde u_0 representa o resultado da desfuzificação, u é a variável de saída e μ a função de pertença.

- **Método da média dos máximos**

Esta técnica fornece a ação de controlo que representa o valor médio de todas as ações de controlo locais cujas funções de pertença correspondem ao valor máximo.

$$u_0 = \sum_{j=1}^l \frac{u_j}{l} \quad (2.36)$$

onde u_0 representa o resultado da desfuzificação, w é a variável de saída e l é o limite máximo de desfuzificação.

- **Método do primeiro dos máximos**

Esta técnica usa a união dos conjuntos difusos e seleciona o menor valor do domínio com máximo grau de pertença. Esta técnica pode ser representada pela seguinte expressão:

$$u_0 = \min \{u \in U\} : \mu(u) = \max(U) \quad (2.37)$$

onde u_0 representa o resultado da desfuzificação e u a variável de saída.

CONTROLADORES DIFUSOS DO TIPO MAMDANI

Os controladores do tipo Proporcional-Integral-Derivativo (PID) convencionais continuam a ser muito utilizados em controlo de processos. A principal motivação para o uso deste tipo de controladores está relacionada com o facto de serem fáceis de sintonizar, terem baixo custo e apresentarem simplicidade funcional e estrutural. Quando esta metodologia lida com sistemas lineares ou mesmo com sistemas dinâmicos ligeiramente não lineares, o controlador PID com ganhos fixos pode fornecer teoricamente um desempenho aceitável em anel fechado, em alargados regimes de operação. Contudo, esta situação não se verifica quando se está perante não linearidades moderadas ou significativas. Nestas condições, a técnica acima descrita poderá conduzir a um baixo desempenho em anel fechado, ou até mesmo à instabilidade. Nas situações em que o sistema é não linear, a lógica difusa fornece as ferramentas para a conceção de estruturas de controlo, em particular os controladores PID difusos do tipo Mamdani [Fen06].

3.1 Controladores PID difusos

Os controladores difusos foram desenvolvidos para emular o desempenho de operadores humanos, através da codificação do seu conhecimento sob a forma de regras linguísticas. Este tipo de controladores proporcionam uma alternativa à metodologia de controlo analítica convencional [God00].

Um controlador difuso PID convencional tem por base uma simples combinação linear de três sinais [Jan07]: P- ação proporcional ao erro, e ; I- ação proporcional ao integral do erro, $\int e \, dt$; D- ação proporcional à derivada no tempo do erro, \dot{e} . Através destas três componentes, a expressão geral, em tempo contínuo, do controlador é representada de

acordo com a seguinte equação [Dri+93]:

$$u(t) = K_P \cdot e(t) + K_I \cdot \int_0^t e(\tau) d\tau + K_D \cdot \dot{e}(t) \quad (3.1)$$

em que K_P , K_I e K_D representam os ganhos do controlador associados às componentes proporcional, integral e derivativa, respetivamente. Para se poder obter a versão incremental do controlador convencional PID deriva-se (3.1), obtendo-se então:

$$\dot{u}(t) = K_P \cdot \dot{e}(t) + K_I \cdot e(t) + K_D \cdot \ddot{e}(t) \quad (3.2)$$

Em relação à versão em tempo discreto da equação (3.2), esta pode ser aproximada pela seguinte expressão:

$$\Delta u(k) = K_P \cdot \Delta e(k) + K_I \cdot e(k) + K_D \cdot \Delta^2 e(k) \quad (3.3)$$

Nas subsecções seguintes serão abordados três tipos de controladores difusos, nomeadamente os controladores PI (Proporcional Integral), PD (Proporcional Derivativo) e PID, que têm por base o método de inferência do tipo Mamdani. É de frisar que existem dois tipos básicos de controladores do tipo Mamdani, designadamente o controlador difuso PI, que gera a saída incremental do controlador e o controlador difuso PD, que disponibiliza a ação de controlo. Através destes dois controladores básicos é possível obter-se o controlador PID, agrupando para isso a saída de cada um deles [Dri+93; LG96].

3.1.1 Controlador PD difuso

O controlador PD difuso utiliza a ação derivativa, a qual contribui para melhorar a estabilidade em anel fechado e diminuir o tempo de resposta em regime transitório. Este controlador adota como variáveis linguísticas de entrada o erro e a variação do erro, e exibe como variável linguística de saída a ação de controlo. A equação que descreve este controlador é expressa por:

$$\tilde{u}^{PD}(k) = f_{PD}(\tilde{e}(k), \Delta \tilde{e}(k)) \quad (3.4)$$

em que $f_{PD}(\cdot)$ representa o mapeamento através da base de regras, que permite obter a variável de saída a partir das variáveis de entrada. Quanto às variáveis \tilde{e} , $\Delta \tilde{e}$ e \tilde{u}^{PD} , estas representam os valores normalizados do erro, variação do erro e ação de controlo, respetivamente. As equações que permitem relacionar os valores normalizados com os valores crespos são:

$$\tilde{e}(k) = K_e \cdot e(k) \quad (3.5)$$

$$\Delta \tilde{e}(k) = K_e \cdot \Delta e(k) \quad (3.6)$$

$$u^{PD}(k) = K_u \cdot \tilde{u}^{PD}(k) \quad (3.7)$$

onde K_e , $K_{\Delta e}$ e K_u representam os fatores de escala do controlador difuso associados ao erro, variação do erro e ação de controlo, respetivamente. É importante salientar que os fatores de escala associados às variáveis de entrada desempenham um papel fulcral na normalização do universo de discurso, enquanto que o fator de escala referente à saída tem por objetivo efetuar a devida desnormalização. A representação da estrutura do controlador PD encontra-se ilustrada na figura 3.1.

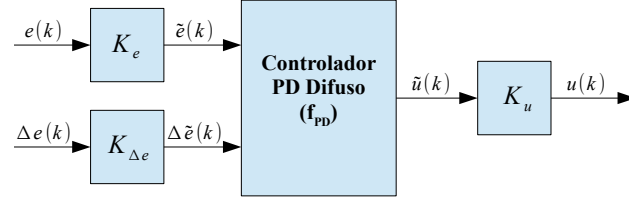


Figura 3.1: Estrutura do controlador PD difuso.

Ainda em relação aos fatores de escala, tanto de entrada como de saída, têm funcionalidades semelhantes aos ganhos do controlador convencional, tendo um efeito significativo no que diz respeito à estabilidade e desempenho do controlador [Rez+00].

Tendo em conta a topologia difusa PD representada na figura 3.1, a base de regras do controlador toma a seguinte forma:

$$\text{Regra } i, j: \text{ Se } \tilde{e} \text{ é } LE_{(i)} \text{ e } \Delta\tilde{e} \text{ é } L\Delta E_{(j)} \text{ então } \tilde{u}^{PD} \text{ é } LU_{(m)}^{PD} \quad (3.8)$$

em que \tilde{e} , $\Delta\tilde{e}$ representam as variáveis linguísticas de entrada e \tilde{u}^{PD} a variável linguística de saída. As variáveis $LE_{(i)}$, com $i = 0, 1, \dots, N_1 - 1$, e $L\Delta E_{(j)}$, com $j = 0, 1, \dots, N_2 - 1$, representam os termos linguísticos dos antecedentes e $LU_{(m)}^{PD}$, com $m = 0, 1, \dots, M - 1$, os termos linguísticos associados à saída do controlador, sendo N_1 e N_2 o número de termos linguísticos relativos às variáveis de entrada, enquanto que M se refere ao número de valores linguísticos atribuído ao consequente. A tabela 3.1 apresenta uma base de regras característica do controlador PD difuso [LG96], composta por quarenta e nove regras padrão.

Tabela 3.1: Base de regras genérica para o controlador PD difuso.

e Δe	NG	NM	NP	ZO	PP	PM	PG
NG	NG	NG	NG	NG	NM	NP	NP
NM	NG	NG	NG	NM	NP	NP	NP
NP	NG	NG	NM	NP	NP	NP	NP
ZO	NP	NP	NP	ZO	PP	PP	PP
PP	PP	PP	PP	PP	PM	PG	PG
PM	PP	PP	PP	PM	PG	PG	PG
PG	PP	PP	PM	PG	PG	PG	PG

3.1.2 Controlador PI difuso

O controlador PI difuso utiliza o efeito integral para anular o erro estático. Este controlador toma como variáveis linguísticas de entrada o erro e a variação do erro e apresenta como variável linguística de saída o incremento da ação de controlo. A equação que descreve a saída deste controlador difuso é obtida com base no cálculo do operador somador, sendo esta descrita por:

$$\Delta \tilde{u}^{PI}(k) = f_{PI}(\tilde{e}(k), \Delta \tilde{e}(k)) \quad (3.9)$$

em que $f_{PI}(\cdot)$ representa o mapeamento através da base de regras, que permite obter a variável de saída a partir das variáveis de entrada. Quanto às variáveis \tilde{e} , $\Delta \tilde{e}$ e $\Delta \tilde{u}^{PI}$, estas representam os valores normalizados do erro, variação do erro e variação da ação de controlo, respetivamente. As equações que permitem relacionar os valores normalizados com os valores de entrada crespos são as mesmas que foram consideradas na secção 3.1.1, nomeadamente, as equações (3.5) e (3.6). Em relação à saída, a relação entre valor normalizado e valor crespo é dada por:

$$\Delta u^{PI}(k) = K_{\Delta u} \cdot \Delta \tilde{u}^{PI}(k) \quad (3.10)$$

em que $K_{\Delta u}$ representa o fator de escala do controlador difuso associado ao incremento da ação de controlo. Assim como no controlador PD, secção 3.1.1, os fatores de escala também desempenham um papel importante no controlador PI, no que diz respeito à normalização do universo de discurso e também na sintonização do controlador. No caso do fator de escala $K_{\Delta u}$, este tem por objetivo a desnormalização do universo de discurso da variável de saída. A representação da estrutura do controlador PI encontra-se ilustrada na figura 3.2.

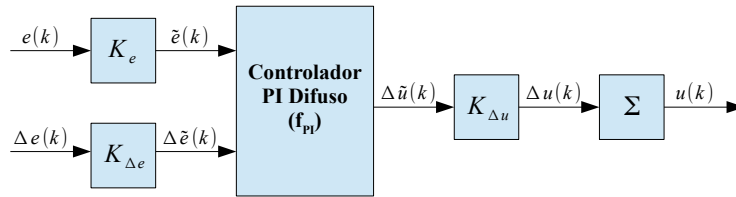


Figura 3.2: Estrutura do controlador PI difuso.

Tendo em conta a topologia representada na figura 3.2, a base de regras do controlador toma a seguinte forma:

$$\text{Regra } i, j: \text{ Se } \tilde{e} \text{ é } LE_{(i)} \text{ e } \Delta \tilde{e} \text{ é } L\Delta E_{(j)} \text{ então } \Delta \tilde{u}^{PI} \text{ é } LU_{(m)}^{PI} \quad (3.11)$$

em que \tilde{e} , $\Delta \tilde{e}$ representam as variáveis linguísticas de entrada e $\Delta \tilde{u}^{PI}$ a variável linguística de saída. As variáveis $LE_{(i)}$, com $i = 0, 1, \dots, N_1 - 1$, e $L\Delta E_{(j)}$, com $j = 0, 1, \dots, N_2 - 1$, representam os termos linguísticos dos antecedentes e $LU_{(m)}^{PI}$, com $m = 0, 1, \dots, M - 1$, os termos linguísticos associados à saída do controlador, sendo N_1 e N_2 o número de termos linguísticos relativos às variáveis de entrada, enquanto que M se refere ao número

de valores linguísticos atribuído ao consequente. A tabela 3.2 apresenta uma base de regras característica do controlador PI difuso [LG96], composta por quarenta e nove regras padrão.

Tabela 3.2: Base de regras genérica para o controlador PI difuso.

$e \Delta e$	NG	NM	NP	ZO	PP	PM	PG
NG	NG	NG	NG	NG	NM	NP	ZO
NM	NG	NG	NG	NM	NP	ZO	PP
NP	NG	NG	NM	NP	ZO	PP	PM
ZO	NG	NM	NP	ZO	PP	PM	PG
PP	NM	NP	ZO	PP	PM	PG	PG
PM	NP	ZO	PP	PM	PG	PG	PG
PG	ZO	PP	PM	PG	PG	PG	PG

O controlador PI difuso apresenta uma configuração semelhante à do controlador PD, i.e, o mecanismo de inferência é análogo em ambos os controladores, encontrando-se as principais diferenças relacionadas com a variável de saída do controlador e base de regras. Em relação à base de regras, a principal diferença entre os controladores PI e PD difusos está relacionada com a linha de comutação, onde o sinal das variáveis de saída muda entre positivo para negativo. A linha de comutação para o controlador PI difuso é diagonal ($E = -\Delta E$), enquanto que no caso do controlador PD difuso a comutação ocorre na linha em que $E = ZO$. As linhas mencionadas encontram-se sombreadas a azul nas tabelas 3.1 e 3.2.

3.1.3 Controlador PID difuso baseado em PI+PD

O controlador PID difuso considera três variáveis linguísticas de entrada, nomeadamente o erro, a variação do erro e a segunda variação do erro, e apresenta como variável linguística de saída o incremento da ação de controlo. A equação que descreve a saída deste controlador difuso é dada por:

$$\Delta \tilde{u}^{PID}(k) = f_{PID}(\tilde{e}(k), \Delta \tilde{e}(k), \Delta^2 \tilde{e}(k)) \quad (3.12)$$

em que $f_{PID}(\cdot)$ representa o mapeamento através da base de regras, que permite obter a variável de saída a partir das variáveis de entrada. Quanto às variáveis \tilde{e} , $\Delta \tilde{e}$, $\Delta^2 \tilde{e}$ e $\Delta \tilde{u}^{PI}$, estas representam os valores normalizados do erro, variação do erro, segunda variação do erro e variação da ação de controlo, respetivamente. As equações que permitem relacionar os valores normalizados com os valores de entrada crespas, idênticas às equações (3.5) e (3.6), incluem uma equação adicional referente à segunda variação do erro. Quanto à relação entre o valor normalizado e o valor cresco da variável de saída, esta é a mesma que foi considerada na secção 3.1.2, designadamente (3.10). A equação que representa a relação mencionada para a segunda variação do erro é expressa por:

$$\Delta^2 \tilde{e}(k) = K_{\Delta^2 e} \cdot \Delta^2 e(k) \quad (3.13)$$

onde $K_{\Delta^2 e}$ representa o fator de escala do controlador difuso associado à variação do erro. A representação do controlador PID encontra-se ilustrada na figura 3.3.

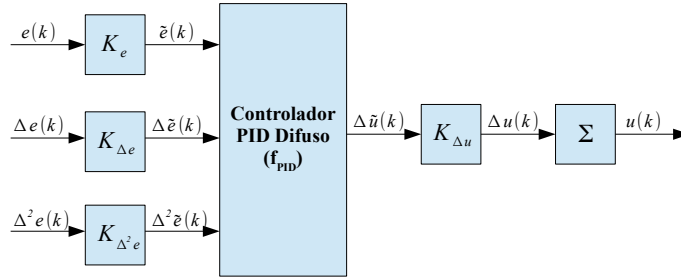


Figura 3.3: Estrutura do controlador PID difuso.

Tendo em conta a figura 3.3, a base de regras do controlador apresenta a seguinte forma:

$$\text{Regra } i, j, l : \text{ Se } \tilde{e} \text{ é } LE_{(i)}, \Delta \tilde{e} \text{ é } L\Delta E_{(j)} \text{ e } \Delta^2 \tilde{e} \text{ é } L\Delta^2 E_{(l)} \text{ então } \Delta \tilde{u}^{PI} \text{ é } LU_{(m)}^{PI} \quad (3.14)$$

em que \tilde{e} , $\Delta \tilde{e}$, $\Delta^2 \tilde{e}$ representam as variáveis linguísticas de entrada e $\Delta \tilde{u}^{PI}$ a variável linguística de saída. As variáveis $LE_{(i)}$, com $i = 0, 1, \dots, N_1 - 1$, $L\Delta E_{(j)}$, com $j = 0, 1, \dots, N_2 - 1$, e $L\Delta^2 E_{(l)}$, com $l = 0, 1, \dots, N_3 - 1$, representam os termos linguísticos dos antecedentes e $LU_{(m)}^{PI}$, com $m = 0, 1, \dots, M - 1$, os termos linguísticos associados à saída do controlador, sendo N_1 , N_2 e N_3 o número de termos linguísticos relativos às variáveis de entrada, enquanto que M se refere ao número de valores linguísticos atribuído ao consequente.

Para este controlador, o facto da regra ser composta por três variáveis linguísticas de entrada, com sete termos linguísticos associados a cada uma, faz com que a base de regras passe a ser composta por $7^3 = 343$ regras. O elevado número de regras torna difícil o projeto e a interpretação do controlador. Por este motivo utiliza-se uma abordagem alternativa [LG96], a qual consiste na combinação das saídas dos controladores PI e PD, ficando assim a versão simplificada do controlador difuso PID com a seguinte forma:

$$u^{PID} = u^{PI}(k) + u^{PD}(k) \quad (3.15)$$

em que u^{PI} e u^{PD} representam as saídas dos controladores PI e PD difusos, respetivamente. A versão simplificada deste controlador reduz a complexidade da base de regras e aumenta a eficiência do controlador, tornando assim mais fácil o projeto e a sua interpretabilidade. A topologia deste controlador encontra-se ilustrada na figura 3.4.

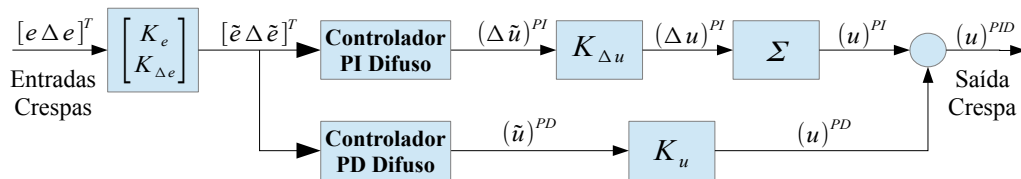


Figura 3.4: Estrutura do controlador PID difuso simplificada.

MODELAÇÃO EXPERIMENTAL DE SISTEMAS

Um sistema pode ser encarado como um "objeto" em que diferentes variáveis interagem, em todo o tipo de escalas de tempo e de espaço, e que produzindo sinais observáveis. Quando o conhecimento prévio do sistema é limitado e a disponibilidade dos dados observados é incompleta, não é possível obter uma descrição matemática exata do sistema. Mesmo que exista um conhecimento total do sistema ou que estejam disponíveis dados suficientes, a descrição exata é na maioria das vezes rejeitada, pois o modelo iria torna-se demasiado complexo para ser usado numa aplicação. Através destes factos pode-se então concluir que um modelo matemático é sempre uma aproximação de um sistema real, sendo esta a base da identificação de sistemas, que é considerada como a modelação aproximada para uma aplicação específica, com base em dados observados e conhecimento prévio do sistema [Kee11].

A identificação de sistemas segue um procedimento composto por quatro etapas [Lju99]: a primeira referente à recolha de dados do sistema, a segunda à escolha da estrutura do modelo, a terceira ao cálculo dos parâmetros do modelo e, por último, à validação do modelo. O problema principal na identificação de sistemas é encontrar uma estrutura de modelos adequada, sendo a estimação do modelo usualmente considerada um problema menor. Uma das regras básicas na estimação é *não estimar aquilo que já se sabe*, ou seja, deve-se usar conhecimento prévio e intuição sobre o comportamento do sistema, quando se escolhe a estrutura do modelo. Existem fundamentalmente três formas distintas de representar o comportamento dinâmico do sistemas: [Sjö+95]:

- **Modelos caixa-branca:** Baseiam-se nos fenómenos físicos que definem o comportamento do sistema;
- **Modelos caixa-cinzenta:** Conhece-se parte do comportamento físico do sistema, mas existem vários parâmetros que necessitam ser determinados. Neste modelo

consideram-se dois sub-casos:

1. Modelação física: A estrutura do modelo pode ser construída a partir de fenómenos físicos, subsistindo um conjunto de parâmetros a ser estimado a partir dos dados.
 2. Modelação semi-física: O conhecimento físico é usado para sugerir algumas combinações não lineares do sinal de dados medido. Estes novos sinais são então submetidos a uma modelação do tipo caixa-preta.
- **Modelos caixa-preta:** Nenhum conhecimento físico está disponível ou é utilizado na modelação do sistema. O modelo é escolhido a partir de um conjunto de estruturas de modelos empíricos.

Em seguida serão abordados os temas relativos à modelação de sistemas lineares e não lineares.

4.1 Modelação linear

O modelo linear mais simples é o modelo FIR (Finite Impulse Response), sendo este representado pela seguinte equação [Sjö+95]:

$$y(k) = B(q^{-1})u(k) + e(k) \quad (4.1)$$

onde q^{-1} denota o operador de atraso e $e(k)$ o ruído. Na prática, as estruturas lineares usadas são variantes da equação (4.1), através da alteração da forma como os pólos são calculados ou como as características do ruído são descritas. A estrutura geral dos modelos lineares (ver Figura 4.1) toma a forma apresentada pela equação (4.2).

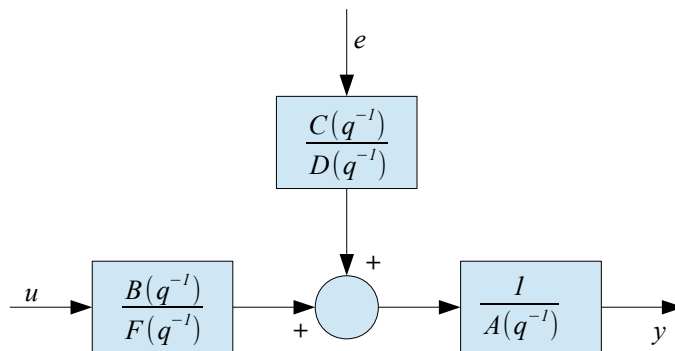


Figura 4.1: Estrutura geral dos modelos lineares.

$$A(q^{-1})y(k) = \frac{B(q^{-1})}{F(q^{-1})}u(k) + \frac{C(q^{-1})}{D(q^{-1})}e(k) \quad (4.2)$$

onde os polinómios $A(q^{-1})$, $B(q^{-1})$, $C(q^{-1})$, $D(q^{-1})$ e $F(q^{-1})$ são expressos por,

$$\begin{aligned} A(q^{-1}) &= 1 + a_1 q^{-1} + \dots + a_{n_a} q^{-n_a} \\ B(q^{-1}) &= b_1 q^{-1} + b_2 q^{-2} + \dots + b_{n_b} q^{-n_b} \\ C(q^{-1}) &= 1 + c_1 q^{-1} + \dots + c_{n_c} q^{-n_c} \\ D(q^{-1}) &= 1 + d_1 q^{-1} + \dots + d_{n_d} q^{-n_d} \\ F(q^{-1}) &= 1 + f_1 q^{-1} + \dots + f_{n_f} q^{-n_f} \end{aligned} \quad (4.3)$$

em que n_a , n_b , n_c , n_d e n_f representam as ordens dos respetivos polinómios. Pela análise da equação (4.2), verifica-se que $A(q^{-1})$ corresponde aos pólos do sistema, comuns tanto à entrada como ao ruído. Os zeros e pólos associados apenas à entrada do sistema encontram-se representados por $B(q^{-1})$ e $F(q^{-1})$, respetivamente, enquanto que os polinómios $C(q^{-1})$ e $D(q^{-1})$ são referentes aos zeros e pólos associados ao ruído.

Dependendo de quais destes cinco polinómios são utilizados, é possível obter trinta e duas diferentes estruturas de modelos lineares [Lju99]. Na tabela 4.1 encontram-se apresentadas cinco das estruturas mais comuns.

Tabela 4.1: Algumas das estruturas mais utilizadas.

Estrutura do modelo ¹	Polinómios usados em (4.2)	Estrutura resultante
ARX	AB	$A(q^{-1}) y(k) = B(q^{-1}) u(k) + e(k)$
ARMAX	ABC	$A(q^{-1}) y(k) = B(q^{-1}) u(k) + C(q^{-1}) e(k)$
ARMA	AC	$A(q^{-1}) y(k) = u(k) + C(q^{-1}) e(k)$
OE	BF	$y(k) = \frac{B(q^{-1})}{F(q^{-1})} u(k) + e(k)$
BJ	BFGD	$y(k) = \frac{B(q^{-1})}{F(q^{-1})} u(k) + \frac{C(q^{-1})}{D(q^{-1})} e(k)$

Sublinha-se que das cinco estruturas de modelos apresentadas na tabela 4.1, a estrutura ARX foi a considerada no âmbito da presente dissertação. Apesar desta estrutura apresentar como desvantagem o facto do ruído se submeter à dinâmica do polinómio $A(q^{-1})$, antes de ser adicionado à saída, possui como alicante o facto da propriedade do preditor definir uma regressão linear.

4.1.1 Estimação em diferido dos parâmetros do modelo ARX

A estimação em diferido dos parâmetros do modelo ARX baseia-se no método dos mínimos quadráticos [Lju99], assumindo o preditor definido pela seguinte regressão linear:

$$\hat{y}(k|\theta) = \varphi^T \theta \quad (4.4)$$

¹ARX=Auto Regressive with eXogenous input; ARMAX=Auto Regressive Moving Average with eXogenous input; ARMA=Auto Regressive Moving Average; OE=Output Error; BJ=Box Jenkins

em que o regressor φ é constituído por entradas e saídas passadas, (4.5), e o vetor de parâmetros θ definido por (4.6):

$$\varphi(k) = [-y(k-1) \cdots -y(k-n_a) \quad u(k-1) \cdots u(k-n_b)]^T \quad (4.5)$$

$$\theta = [a_1 \quad a_2 \cdots a_{n_a} \quad b_1 \quad b_2 \cdots b_{n_b}]^T \quad (4.6)$$

Os pares $\{y(k), u(k)\}$, com $k = 1, \dots, N$, são obtidos através de ensaios sobre o sistema a identificar, enquanto que o vetor de parâmetros θ , é escolhido de forma a minimizar a diferença entre a saída do modelo (preditor) e a saída do sistema.

O método dos mínimos quadráticos consiste na determinação dos parâmetros θ , de forma a minimizar o seguinte critério:

$$V_N(\theta, Z^N) = \frac{1}{N} \sum_{k=1}^N (\varepsilon(k, \theta))^2 \quad (4.7)$$

em que Z^N representa os dados de entrada e saída do sistema e $\varepsilon(k, \theta)$ o erro de modelação, ou seja,

$$Z^N = \{y(k), u(k)\}, \text{ com } k = 1, \dots, N \quad (4.8)$$

$$\varepsilon(k, \theta) = y(k) - \hat{y}(k|\theta) = y(k) - \varphi^T(k) \theta \quad (4.9)$$

O vetor de parâmetros θ é escolhido de forma a minimizar (4.7), ou seja,

$$\hat{\theta}_N = \arg \min_{\theta} V_N(\theta, Z^N) \quad (4.10)$$

onde $\hat{\theta}_N$ representa o valor de θ que minimiza V_N .

4.1.2 Estimação recursiva dos parâmetros do modelo ARX

A estimação recursiva dos parâmetros do modelo ARX tem por base o método dos mínimos quadráticos recursivos [Lju99], pois na identificação em-linha as observações surgem sequencialmente em tempo real. Considerando um estimador $\hat{\theta}$ dado por:

$$\hat{\theta}(k) = P(k) \sum_{s=1}^k \varphi(s) y(s) \quad (4.11)$$

em que $\hat{\theta}(k)$ representa uma estimativa no instante k do vetor de parametrização e $P(k)$ a matriz de covariância. Desenvolvendo a equação (4.11) resulta:

$$\begin{aligned} \hat{\theta}(k) &= P(k) \left(\sum_{s=1}^{k-1} \varphi(s) y(s) + \varphi(k) y(k) \right) \Rightarrow \\ \Rightarrow P^{-1}(k) \hat{\theta}(k) &= \sum_{s=1}^{k-1} (\varphi(s) y(s)) + \varphi(k) y(k) \end{aligned} \quad (4.12)$$

obtendo-se a seguinte relação:

$$P^{-1}(k) \hat{\theta}(k) = P^{-1}(k-1) \hat{\theta}(k-1) + \varphi(k) y(k) \quad (4.13)$$

Uma vez que $P^{-1}(k)$ é dado por:

$$P^{-1}(k) = \sum_{s=1}^k \varphi(s) \varphi^T(s) \quad (4.14)$$

resulta,

$$\begin{aligned} P^{-1}(k) &= \sum_{s=1}^k \varphi(s) \varphi^T(s) \\ &= \sum_{s=1}^{k-1} \left(\varphi(s) \varphi^T(s) \right) + \varphi(k) \varphi^T(k) \\ &= P^{-1}(k-1) + \varphi(k) \varphi^T(k) \end{aligned} \quad (4.15)$$

pelo que,

$$P(k) = \left[P^{-1}(k-1) + \varphi(k) \varphi^T(k) \right]^{-1} \quad (4.16)$$

A equação (4.16) pode ser transformada com base no lema da inversão de matrizes,

$$(A + BCD)^{-1} = A^{-1} - A^{-1}B \left(C^{-1} + DA^{-1}B \right)^{-1} DA^{-1} \quad (4.17)$$

considerando $A = P^{-1}(k-1)$, $B = \varphi(k)$, $C = 1$ e $D = \varphi^T(k)$ obtém-se,

$$P(k) = P(k-1) \left[I_n - \frac{\varphi(k) \varphi^T(k) P(k-1)}{1 + \varphi^T(k) P(k-1) \varphi(k)} \right] \quad (4.18)$$

onde I_n representa a matriz identidade com as dimensões de $P(k)$. De modo a tornar o método dos mínimos quadráticos recursivos adaptável a pequenas mudanças no sistema, como por exemplo desgaste de alguns componentes, é incorporado um fator de esquecimento λ a (4.18). A expressão final da matriz de covariância $P(k)$ toma a seguinte forma:

$$P(k) = \frac{P(k-1)}{\lambda} \left[I_n - \frac{\varphi(k) \varphi^T(k) P(k-1)}{\lambda + \varphi^T(k) P(k-1) \varphi(k)} \right] \quad (4.19)$$

com λ normalmente compreendido entre 0,95 e 1.

A equação (4.15) pode ser reescrita da seguinte forma:

$$P^{-1}(k-1) = P^{-1}(k) - \varphi(k) \varphi^T(k) \quad (4.20)$$

Substituindo a expressão (4.20) em (4.13) obtém-se a atualização recursiva do vetor de parâmetros:

$$\hat{\theta}(k) = \hat{\theta}(k-1) + P(k) \varphi(k) \varepsilon(k) \quad (4.21)$$

em que $\varepsilon(k)$ corresponde ao erro do modelação, ou seja,

$$\varepsilon(k) = y(k) - \hat{y}(k|\hat{\theta}(k-1)) = y(k) - \varphi^T(k) \hat{\theta}(k-1) \quad (4.22)$$

4.2 Modelação não linear

À semelhança da modelação linear, também na modelação não linear existem vários tipos de estruturas de modelos. Algumas destas arquiteturas baseiam-se em modificações das estruturas mencionadas na secção 4.1, tais como NARX, NARMA ou NOE², entre outras.

No presente trabalho foi considerado o modelo não linear baseado na estrutura NNARX (Neural Network Auto Regressive with eXogenous input), que é um subconjunto da estrutura NARX [Rah+08] baseada em redes neuronais. As redes neuronais têm sido bastante utilizadas no contexto de sistemas não lineares, especialmente em identificação e controlo. Este facto deve-se principalmente à sua capacidade inerente de aprender o mapeamento multidimensional não linear de entrada-saída, sem qualquer tipo de conhecimento sobre a estrutura do sistema ou a sua dinâmica. Matematicamente o modelo de um neurónio (ver Figura 4.2) pode ser descrito pela seguinte expressão [Kis08]:

$$\begin{cases} x = \sum_{i=1}^n w_i u_i + b \\ N_0 = f(x) \end{cases} \quad (4.23)$$

onde x , $u_i (i = 1, \dots, n)$ e $w_i (i = 1, \dots, n)$ correspondem à soma ponderada das entradas, entradas e pesos, respetivamente. A saída do neurónio é representado por N_0 e a função de ativação por f .

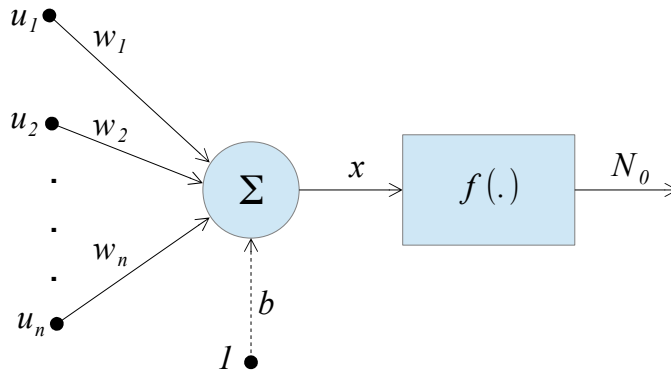


Figura 4.2: Estrutura de um neurónio.

Numa rede neuronal do tipo NNARX, a relação entrada-saída é representada por [Rah+08]:

$$\hat{y}(k) = g(\varphi(k), W) \quad (4.24)$$

em que $g(\cdot)$ representa o mapeamento não linear da rede neuronal, $\varphi(k)$ o vetor de regressão no instante k , composto por entradas e saídas passadas, W a matriz de pesos e $\hat{y}(k)$ o preditor no instante k .

²NARX=Nonlinear Auto Regressive with eXogenous input; NARMA=Nonlinear Auto Regressive Moving Average; NOE=Nonlinear Output Error

O esquema de blocos da estrutura NNARX encontra-se ilustrado na figura 4.3, onde o vetor de regressão toma a seguinte forma:

$$\varphi(k) = [y(k-1) \cdots y(k-n_a) \ u(k-1) \cdots u(k-n_b)]^T \quad (4.25)$$

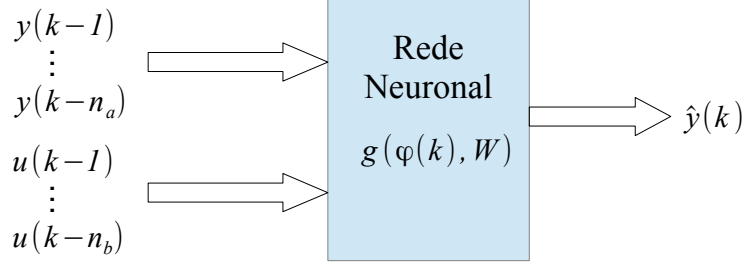


Figura 4.3: Estrutura de uma rede neuronal NNARX.

4.2.1 Estimação em diferido dos parâmetros da rede neuronal

A estimação em diferido dos parâmetros do modelo NNARX pode ser efetuada, por exemplo através do algoritmo de retropropagação, sendo este um dos mais usados. Na essência, este algoritmo consiste num procedimento estocástico com base no gradiente descendente que visa minimizar o critério do erro quadrático [Liu02].

O algoritmo de retropropagação será aplicado a uma rede com $M + 1$ camadas representado na figura 4.4.

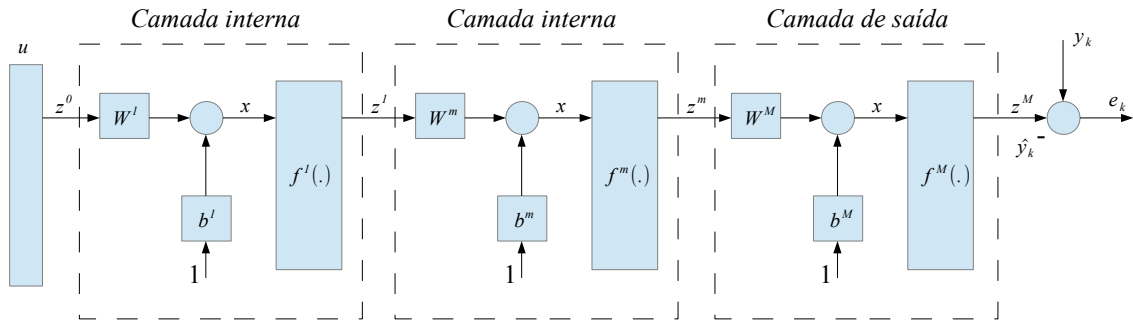


Figura 4.4: Rede neuronal composta por $M+1$ camadas.

Este algoritmo [Gra07] parte da camada de saída, pois esta é a única onde a saída está disponível. Definindo o critério do erro quadrático $J(\cdot)$ como:

$$J(\cdot) = \frac{1}{2} \sum_k e_k^2 = \frac{1}{2} \sum_k (y_k - \hat{y}_k)^2 \quad (4.26)$$

onde $k = 1, \dots, N$, com N sendo o número de neurónios da camada de saída, y_k a saída real do sistema, associada ao neurónio k , e \hat{y}_k a saída da rede neuronal (preditor),

associada ao neurónio k . A atualização dos pesos, w , e das polaridades, b , da rede neuronal, associados à camada m , é efetuada da seguinte forma:

$$\begin{cases} w_{kj}^m(l+1) = w_{kj}^m(l) + \Delta w_{kj}^m(l) \\ b_k^m(l+1) = b_k^m(l) + \Delta b_k^m(l) \end{cases} \quad (4.27)$$

em que l representa o conjunto de treino atual; $m = 1, \dots, M$ a camada da rede; k e j os índices associados ao neurónio e à entrada da camada m , respetivamente. Através do método do gradiente descendente, o cálculo da variação dos pesos e das polaridades toma a seguinte forma:

$$\begin{cases} \Delta w_{kj}^m = -\eta \nabla J|_{w_{kj}^m} = -\eta \frac{\partial J}{\partial w_{kj}^m} \\ \Delta b_k^m = -\eta \nabla J|_{b_k^m} = -\eta \frac{\partial J}{\partial b_k^m} \end{cases} \quad (4.28)$$

onde η representa um fator de escala e o sinal $(-)$ indica a direção descendente em direção a um mínimo.

Como o critério do erro não é representado por um função explícita dos pesos e polaridades associados às camadas internas, é necessário recorrer-se à regra da cadeia para o cálculo das respetivas derivadas. Através do uso desta regra as derivadas são expressas por:

$$\begin{cases} \frac{\partial J}{\partial w_{kj}^m} = \frac{\partial J}{\partial x_k^m} \cdot \frac{\partial x_k^m}{\partial w_{kj}^m} \\ \frac{\partial J}{\partial b_k^m} = \frac{\partial J}{\partial x_k^m} \cdot \frac{\partial x_k^m}{\partial b_k^m} \end{cases} \quad (4.29)$$

Pela análise da figura 4.4, retira-se que:

$$x^m = W^m z^{m-1} + b^m \Rightarrow x_k^m = \sum_j w_{kj}^m \cdot z_j^{m-1} + b_k^m \quad (4.30)$$

logo,

$$\begin{cases} \frac{\partial x_k^m}{\partial w_{kj}^m} = z_j^{m-1} \\ \frac{\partial x_k^m}{\partial b_k^m} = 1 \end{cases} \quad (4.31)$$

Definindo:

$$\Phi_k^m = -\frac{\partial J}{\partial x_k^m} \quad (4.32)$$

e substituindo (4.31) e (4.32) em (4.29) e o resultado em (4.28) obtém-se:

$$\begin{cases} \Delta w_{kj}^m = \eta \Phi_k^m \cdot z_j^{m-1} \\ \Delta b_k^m = \eta \Phi_k^m \end{cases} \quad (4.33)$$

Aplicando novamente a regra da cadeia a (4.32) resulta:

$$\Phi_k^m = -\frac{\partial J}{\partial x_k^m} = -\frac{\partial J}{\partial z_k^m} \cdot \frac{\partial z_k^m}{\partial x_k^m} \quad (4.34)$$

Através da figura 4.4 verifica-se:

$$z^m = f(x_k^m) \quad (4.35)$$

pelo que,

$$\frac{\partial z_k^m}{\partial x_k^m} = \dot{f}(x_k^m) \quad (4.36)$$

onde \dot{f} representa a derivada da função de ativação f . A obtenção da equação (4.34) requer o cálculo de uma outra derivada, sendo necessário aplicar de novo a regra da cadeia:

$$\frac{\partial J}{\partial z_k^m} = \sum_i \frac{\partial J}{\partial x_i^{m+1}} \cdot \frac{\partial x_i^{m+1}}{\partial z_k^m} \quad (4.37)$$

em que i representa o índice do ramo que liga o neurónio k da camada m a um dos neurónios da camada $m + 1$. A partir de (4.30) vem:

$$\frac{\partial x_i^{m+1}}{\partial z_k^m} = w_{ik}^{m+1} \quad (4.38)$$

Substituindo (4.32) e (4.38) na expressão (4.37) e substituindo o resultado em (4.34), juntamente com (4.36), resulta:

$$\Phi_k^m = \dot{f}(x_k^m) \cdot \sum_i \Phi_i^{m+1} \cdot w_{ik}^{m+1} \quad (4.39)$$

Finalmente, para o caso da camada de saída, com $m = M$ e $z_k^M = \hat{y}_k$, a equação (4.37) toma a seguinte forma:

$$\frac{\partial J}{\partial \hat{y}_k} = -(y_k - \hat{y}_k) \quad (4.40)$$

Substituindo (4.40) em (4.34) obtém-se:

$$\Phi_k^M = -\dot{f}(x_k^M) (y(k) - \hat{y}(k)) \quad (4.41)$$

4.2.2 Estimação em-linha dos parâmetros do modelo NNARX

A estimação em-linha dos parâmetros do modelo NNARX pode ser obtida, por exemplo, através do filtro de Kalman estendido. Este corresponde a um problema de otimização onde o critério a minimizar é dado por:

$$J(\Theta) = \sum_{k=1}^N [y_k - \hat{y}_k]^T V_k^{-1} [y_k - \hat{y}_k] \quad (4.42)$$

em que $J(\Theta)$ representa o critério de erro quadrático, com Θ correspondente à agregação dos pesos e polaridades da rede neuronal, $k = 1, \dots, N$, com N o número de neurónios da camada de saída, y_k a saída real do sistema, \hat{y}_k a saída da rede neuronal (preditor),

associada ao neurónio k e V_k corresponde à variância V . Considere-se então o seguinte sistema [Rib04]:

$$\begin{cases} \Theta_{k+1} = \Theta_k + w_k \\ \hat{y}_k = h(\Theta_k, U_k) + v_k \end{cases} \quad (4.43)$$

onde $h(\Theta_k, U_k)$ corresponde à observação não linear do modelo, com U_k o vetor de entrada da rede neuronal, e w_k e v_k as variáveis aleatórias gaussianas com variâncias $W(k)$ e $V(k)$. Caso o ruído apresente variância constante V , o seu efeito pode ser omitido. Quanto à variância W , esta afeta a taxa de convergência do algoritmo e o desempenho de seguimento.

A metodologia referente ao filtro de Kalman encontra-se apresentada no Algoritmo 1, onde P representa a matriz de covariância, K o ganho do filtro, $C = \left(\frac{\partial h(\cdot)}{\partial \Theta} \right)_{|\hat{\Theta}(k|k-1)}$ a matriz Jacobiana de h e I a matriz identidade.

Algorithm 1 Filtro de Kalman Estendido

Predição Temporal

$$\Theta(k|k-1) = \hat{\Theta}(k-1|k-1)$$

$$P(k|k-1) = P(k-1|k-1) + W(k-1)$$

Correção

$$K(k) = P(k|k-1) C^T(k) [C(k) P(k|k-1) C^T(k) + V(k)]^{-1}$$

$$\hat{\Theta}(k|k) = \hat{\Theta}(k|k-1) + K(k) (y(k) - \hat{y}(k))$$

$$P(k|k) = (I - K(k) C(k)) P(k|k-1)$$

4.3 Validação de modelos

A validação de modelos [Lju99] tem como objetivo avaliar se um dado modelo representa de forma adequada o comportamento dinâmico do sistema a identificar. Duas das técnicas utilizadas na validação de modelos são a validação cruzada e a análise de resíduos.

4.3.1 Validação cruzada

Este método tem por base a comparação, segundo um critério $V(\cdot)$, do desempenho do modelo, para um dado conjunto de dados diferente do usado na estimação dos parâmetros. Simulando o comportamento do sistema através de um conjunto de validação e aplicando, por exemplo, a métrica dada por,

$$V(\theta|Z^N) = \frac{1}{N} \sum_{k=1}^N [y(k) - \hat{y}(k|\hat{\theta})]^2 \quad (4.44)$$

estabelece-se:

1. Se o somatório dos erros quadráticos for da ordem de grandeza do obtido com base no conjunto de estimação, então o modelo poderá ser considerado adequado;

2. Caso a situação descrita no primeiro ponto não se verifique ter-se-á que procurar outro modelo, alterando a ordem dos polinómios ou selecionado outra estrutura.

4.3.2 Teste dos resíduos

Os resíduos entre saídas do sistema e do modelo deverão apresentar-se como ruído branco e serem independentes da entrada para que o modelo descreva de forma adequada o sistema.

Análise do erro de estimação

Seja a expressão de cálculo dos resíduos dada por:

$$\varepsilon(k, \hat{\theta}_N) = y(k) - \hat{y}(k|\hat{\theta}_N) \quad (4.45)$$

e as expressões da covariância e covariância normalizada definidas, respetivamente, por:

$$\hat{R}_\varepsilon(\tau) = \frac{1}{N} \sum_{k=1}^{N-\tau} \varepsilon(k) \varepsilon(k+\tau) \quad (4.46)$$

$$\tilde{R}_\varepsilon(\tau) = \frac{\hat{R}_\varepsilon(\tau)}{\hat{R}_\varepsilon(0)} \quad (4.47)$$

a condição a que os resíduos deverão obedecer para que possam ser considerados uma sucessão de variáveis aleatórias independentes é representada pela seguinte expressão:

$$|\tilde{R}_\varepsilon(\tau)| \leq \frac{1,96}{\sqrt{N}}, \quad \tau \geq 1 \quad (4.48)$$

tendo em conta um nível de confiança de 95%.

Independência entre resíduos e entradas passadas

Este teste avalia se existe informação contida na saída, originada pela entrada $u(k-\tau)$, que não é reproduzida de forma correta pelo modelo. A expressão da covariância para este teste é dada por:

$$\hat{R}_{\varepsilon u}(\tau) = \frac{1}{N} \sum_{k=\tau+1}^N \varepsilon(k) \tilde{u}(k-\tau) \quad (4.49)$$

onde,

$$\tilde{u}(k) = u(k) - \bar{u}(k) \quad (4.50)$$

Em relação à expressão de covariância normalizada, esta toma a seguinte forma:

$$\tilde{R}_{\varepsilon u}(\tau) \leq \frac{\hat{R}_{\varepsilon u}(\tau)}{\sqrt{\hat{R}_\varepsilon(0) \hat{R}_u(0)}} \quad (4.51)$$

sendo a condição prática de aceitação dada por:

$$\left| \bar{\hat{R}}_{\varepsilon u}(\tau) \right| \leq \frac{1,96}{\sqrt{N}}, \tau \geq 1 \quad (4.52)$$

De sublinhar que a existência de uma correlação entre resíduos e entradas futuras, quando τ é negativo, é uma indicação de que se está perante uma retroação, não implicando a rejeição do modelo.

TÉCNICAS DE OTIMIZAÇÃO

Os métodos de otimização são considerados uma ferramenta de grande importância em engenharia. O conceito de otimização é fundamental na análise de muitas decisões complexas, onde a ideia básica subjacente se traduz na tentativa de conseguir obter a melhor "decisão" possível para um problema. Genericamente um problema de otimização consiste na procura de um máximo ou mínimo, ou seja, de um extremo de uma função, denominada por função de custo, podendo esta ser sujeita a restrições [Gri+09; LY08].

Tendo em conta o tipo de função de custo, os métodos de otimização, do problema de otimização, podem ser divididos em dois grandes grupos: os métodos lineares e os métodos não lineares. O primeiro grupo tem por base a otimização de uma função linear sujeita a restrições lineares. Em relação ao segundo grupo, este é utilizado sobretudo em problemas complexos em que a função de custo apresenta características não lineares [Gri+09].

Problema de otimização

O problema de otimização pode ser definido da seguinte forma [Gri+09]:

1. Especificação de uma estimativa da solução x_0
2. Para $k = 0, 1, \dots$
 - i. Se x_k for uma solução ótima, parar.
 - ii. Determinação da *direção de procura* p_k .
 - iii. Determinação do *comprimento de passo* α_k .
 - iv. Minimização de uma função de custo f através de α_k , com direção p_k :
 $\text{minimização } f(x_k + \alpha_k p_k)$
 $\alpha_k \geq 0$
 - v. Definição de $x_{k+1} = x_k + \alpha_k p_k$ como sendo uma estimativa melhorada da solução.

Como se pode verificar através do problema de otimização atrás descrito, a componente p_k representa a *direção de busca* que "melhora" a solução num certo sentido. O escalar α_k é referente ao *comprimento de passo* que determina o ponto x_{k+1} . Considerando que p_k está disponível e que toma como *direção de procura* uma *direção descendente*, o *comprimento de passo* tem de cumprir a restrição $\alpha_k \geq 0$, devido à direção associada a p_k . O escalar α_k é calculado de forma a "diminuir suficientemente" o valor da função de custo f com direção p_k .

5.1 Ótimo local e ótimo global

O objetivo da otimização baseia-se na procura de um ponto ótimo ou extremo onde a função de custo é máxima ou mínima, dependendo da busca. No entanto, muitas das funções apresentam vários máximos ou mínimos locais, sendo este um dos desafios mais complicados associados ao problema de otimização [Zon06].

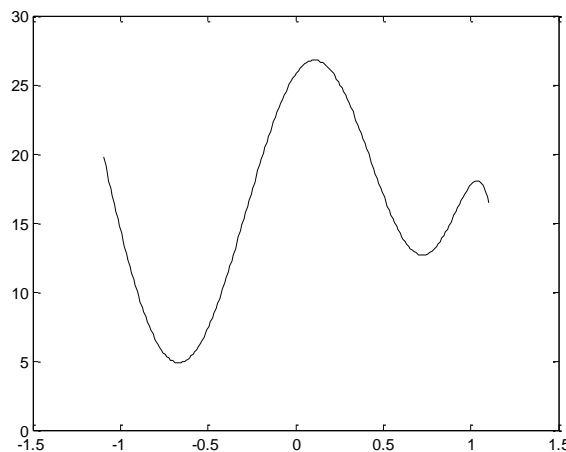


Figura 5.1: Exemplo de uma função com vários extremos.

Admitindo que a busca está direcionada para um ponto que minimize uma determinada função de custo, as definições formais do mínimo local e do mínimo global são dadas por [Gri+09]:

Mínimo local

Um ponto x^* é um ótimo local se existe $\varepsilon > 0$, tal que $f(x^*) \leq f(x)$ para todo o $x \in \mathbb{R}^n$ tal que $\|x - x^*\| < \varepsilon$.

Mínimo global

Um ponto x^* é um ótimo global se existe $f(x^*) \leq f(x)$ para todo o $x \in \mathbb{R}^n$.

De referir que os métodos de busca (algoritmos) ficam muitas vezes "presos" em extremos locais, não sendo possível encontrar o ótimo global. Esta situação encontra-se ilustrada na

figura 5.1 que representa uma função com vários extremos (máximos e mínimos) locais. Um dos aspetos fundamentais referentes ao problema de otimização está relacionado com a convexidade. Um problema de otimização convexo pode ser definido como:

$$\underset{x \in S}{\text{minimização}} f(x) \quad (5.1)$$

em que S representa um conjunto convexo e f uma função de custo convexa contida em S . Considerando x^* um ponto ótimo de (5.1), e dado que, como já foi referido, f representa uma função convexa, pode afirmar-se que x^* corresponde a um mínimo global [Gri+09].

5.2 Métodos de otimização

Os métodos de otimização podem ser determinísticos ou estocásticos [Sar10].

Métodos determinísticos

Caso seja possível prever todas as etapas do problema e ter conhecimento do ponto de partida, tal situação enquadra-se nos métodos determinísticos. De entre este tipo de métodos refiram-se os métodos do tipo gradiente, o método de Newton ou o método de Levenberg-Marquardt, entre outros.

Métodos estocásticos

Neste tipo de métodos são realizadas várias escolhas aleatórias sorteadas no momento da execução do código. Dado que a cada execução do código os números sorteados são diferentes, dificilmente executará a mesma sequência de operações no decurso do processo. Assim, partindo do mesmo ponto inicial cada execução do código seguirá o seu próprio trajeto, e possivelmente obter-se-á uma resposta final diferente. De entre este tipo de algoritmos refiram-se os algoritmos genéticos e os algoritmos evolutivos, entre outros.

5.2.1 Otimização sem restrições

Como nem sempre é possível minimizar funções compostas por múltiplas variáveis, através de abordagens analíticas, recorre-se aos métodos numéricos, os quais têm por base mecanismos com buscas direcionais [SSJ+08].

Método analítico de primeira ordem

Considera-se f uma função de valores reais composta por n variáveis,

$$f(x) = f(x_1, x_2, \dots, x_n) \quad (5.2)$$

O vetor das primeiras derivadas de f é designado por gradiente, ou seja, o gradiente é o vetor das derivadas parciais de f num ponto x , e representa-se por [Gri+09]:

$$\nabla f(x) = \left(\frac{\partial f(x)}{\partial x_1}, \frac{\partial f(x)}{\partial x_2}, \dots, \frac{\partial f(x)}{\partial x_n} \right)^T \quad (5.3)$$

Um dos métodos clássicos de primeira ordem é o método de descida acentuada máxima, expresso por:

$$x^{k+1} = x^k - \alpha_k \nabla f(x^k) \quad (5.4)$$

onde o escalar α_k representa o comprimento de passo. A função de custo ou mérito que minimiza (5.4) é dada por:

$$g_k(\alpha_k) = f(x^k - \alpha_k \nabla f(x^k)), \alpha > 0 \quad (5.5)$$

Este método é descrito no Algoritmo 2.

Algorithm 2 Método de descida acentuada máxima

- 1) Escolher um ponto inicial $x^0, k \leftarrow 0$
 - 2) Calcular a direção de busca $d^k \leftarrow -\nabla f(x^k)$
 - 3) Encontrar α_k tal que $g_k(\alpha_k) = \min_{\alpha > 0} g_k(\alpha)$
 - 4) Calcular $x^{k+1} \leftarrow x^k + \alpha_k d^k$
 - 5) Se o critério de convergência não foi satisfeito, então $k \leftarrow k + 1$
 - 6) Fim.
-

Método analítico de segunda ordem

Um dos métodos analíticos de segunda ordem utilizados na otimização sem restrições é o método de Newton, que apresenta uma estrutura semelhante à do método descrito anteriormente, com a principal diferença da inclusão de uma matriz Hessiana inversa.

A matriz Hessiana é obtida a partir do gradiente, recorrendo-se à segunda derivada da equação (5.3), ou seja:

$$\nabla^2 f(x) = \begin{pmatrix} \frac{\partial^2 f}{\partial x_1 \partial x_1}(x) & \dots & \frac{\partial^2 f}{\partial x_1 \partial x_n}(x) \\ \vdots & \ddots & \vdots \\ \frac{\partial^2 f}{\partial x_n \partial x_1}(x) & \dots & \frac{\partial^2 f}{\partial x_n \partial x_n}(x) \end{pmatrix} \quad (5.6)$$

O método de Newton [Gri+09] considera, normalmente, um comprimento de passo fixo, ou seja, $\alpha_k = 1$, sendo a expressão deste método dada por:

$$x^{k+1} = x^k - \left(\nabla^2 f(x^k) \right)^{-1} \nabla f(x^k) \quad (5.7)$$

O algoritmo referente a este método tem a mesma configuração do correspondente ao método de primeira ordem, com a diferença de que a direção de busca d^k inclui a matriz

inversa da Hessiana, ou seja, $d^k = -(\nabla^2 f(x^k))^{-1} \nabla f(x^k)$ e o comprimento de passo toma um valor de $\alpha_k = 1$.

Uma forma de garantir que a matriz Hessiana seja sempre positiva é através da modificação de Levenberg-Marquardt [Ran04], que adiciona um fator ajustável na diagonal da matriz Hessiana, ou na sua inversa. Como o método de Newton recorre à matriz inversa da Hessiana, a sua expressão toma a seguinte forma:

$$(\nabla^2 f(x^k))^{-1} = (\nabla^2 f(x^k))^{-1} + \lambda I \quad (5.8)$$

em que λ representa um fator de ajuste de escala, designado de coeficiente de regularização, e I uma matriz identidade. Substituindo (5.8) em (5.7) resulta:

$$x^{k+1} = x^k - \left((\nabla^2 f(x^k))^{-1} + \lambda I \right) \nabla f(x^k) \quad (5.9)$$

5.2.2 Otimização com restrições

Um problema de otimização com restrições consiste em minimizar uma função de custo sujeita a restrições de igualdade e/ou desigualdade, em que tanto a função de custo como as restrições podem ser lineares ou não lineares [SSJ+08].

Programação Quadrática Sequencial

O algoritmo de Programação Quadrática Sequencial (PQS) é um dos algoritmos preferidos na otimização não linear com restrições [BT00]. A ideia básica deste método traduz-se na resolução de problemas não lineares com restrições usando, para isso, uma sequência de subproblemas de Programação Quadrática (PQ). As restrições de cada subproblema de PQ representam linearizações de restrições do problema original, e a função de custo do subproblema é uma aproximação quadrática da função Lagrangeana.

Considere-se o problema genérico de programação não linear dado por [Gil+05]:

$$\begin{aligned} & \underset{x}{\text{minimização}} \ f(x) \\ & \text{sujeito a } c(x) \geq 0 \end{aligned} \quad (5.10)$$

em que $x \in \mathbb{R}^n$, $c \in \mathbb{R}^m$, as funções $f(x)$ e $c_i(x)$ têm segundas derivadas contínuas e assume-se que o ponto ótimo (x^*, π^*) existe para (5.10), satisfazendo as condições Karush-Kuhn-Tucker (KKT) de primeira ordem:

$$c(x^*) \geq 0; \ \pi^* \geq 0; \ (c(x^*))^T \pi^* = 0; \ (J(x^*))^T \pi^* = \nabla f(x^*). \quad (5.11)$$

onde π representa os multiplicadores de Lagrange e J é a matriz Jacobiana formada pelos gradientes de cada elemento de c .

A estrutura básica da PQS envolve dois tipos de iterações, nomeadamente as *principais* e as *secundárias*. As iterações principais geram uma sequência de iteradas (x_k, π_k) que convergem para (x^*, π^*) , onde, em cada iteração, se recorre a um subproblema de PQ para obter uma direção de busca em direção a (x_{k+1}, π_{k+1}) . A resolução do subproblema de PQ é em si um procedimento iterativo, onde as iterações secundárias do método de PQS correspondem às iterações do método de PQ. Sendo o Lagrangeano do problema (5.10) definido por [BT00]:

$$L(x, \pi) = f(x) - \pi^T c(x) \quad (5.12)$$

o subproblema de PQ pode ser formulado do seguinte modo [Zhu05]:

$$\begin{aligned} & \underset{d}{\text{minimização}} \quad f(x) + (\nabla f(x_k))^T d + \frac{1}{2} d^T \nabla_{xx}^2 L(x_k, \pi_k) d \\ & \text{sujeito a} \quad c(x_k) + (\nabla c(x_k))^T \cdot d \geq 0 \end{aligned} \quad (5.13)$$

em que d representa a direção de busca.

Neste contexto refira-se o método de PQS sugerido por Wilson [Wil63], bem como SNOPT [Gil+05], LSSQP [Eld92] ou filter-SQP [FL98]¹, entre outros.

Método evolutivo diferencial

Recentemente têm vindo a ser aplicados Algoritmos Evolutivos (AE) na resolução de problemas de engenharia. Os AE são conhecidos pela sua capacidade de lidar com problemas de otimização não linear complexos. A sua principal vantagem face aos outros métodos reside no facto de os AE apenas necessitarem de valores da função de custo e não precisarem de ter em conta propriedades como a diferenciabilidade e continuidade [Mal+11]. Um exemplo deste tipo de algoritmos é o Algoritmo Evolutivo Diferencial (AED), desenvolvido por Price e Storn [SP95].

O AED é um algoritmo evolutivo muito eficaz na otimização global de problemas e é, também, considerado um dos mais eficientes e interessantes em termos das técnicas de computação evolutiva, pois apresenta uma velocidade de convergência elevada e tem poucos parâmetros de controlo em comparação com outros algoritmos evolutivos [BM07; Pis+11]. Este método tem sido aplicado em vários domínios [Mal+11], tais como em problemas de engenharia mecânica [JS99], reconhecimento de padrões [Ilo+03] e processamento de sinais [Sto96], entre outros. Trata-se de um método simples, rápido e robusto que tem um bom desempenho numa ampla variedade de problemas, tendo por base uma técnica de busca estocástica, inerentemente paralela, baseada em populações. A configuração básica do AED encontra-se ilustrada na figura 5.2.

¹SNOPT=Sparse Nonlinear Optimizer; LSSQP=Large Scale Sequential Quadratic Programming; filter-SQP=filter-Sequential Quadratic Programming

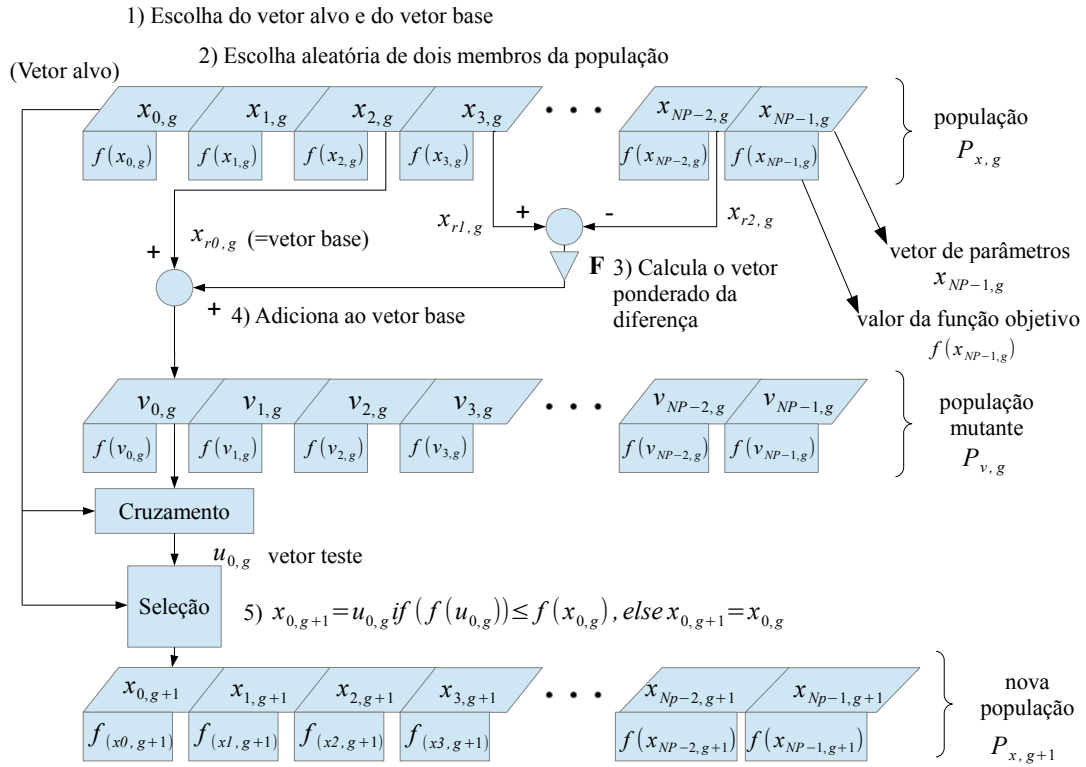


Figura 5.2: Configuração básica do Algoritmo Evolutivo Diferencial.

Este algoritmo baseia-se num método paralelo de procura direta, que usa NP vetores de parâmetros $x_{i,g}$, com $i = 0, 1, \dots, NP - 1$, como população, com dimensão D , para cada geração g , sendo NP o número de membros da população, que é mantido constante ao longo do processo de minimização. A população de vetores inicial é escolhida aleatoriamente devendo abranger todo o espaço de parâmetros. No processo de otimização o AED gera novos vetores de parâmetros através da adição da diferença ponderada entre dois vetores da população a um terceiro vetor. Esta operação é denominada de mutação e os parâmetros subjacentes deste vetor mutado são subsequentemente misturados com os parâmetros de outro vetor pré-determinado (cruzamento), nomeadamente o vetor alvo, por forma a obter o vetor de teste. Se o vetor de teste apresentar um valor de função de custo mais baixo do que o vetor alvo, então o vetor de teste irá substituir o vetor alvo na próxima geração, sendo esta operação denominada de seleção. Cada vetor da população tem de ser usado uma vez como vetor alvo de forma a que ocorram NP competições numa geração [SP95]. Em relação ao critério de paragem deste algoritmo, este é atingido através de um tempo limite pré-especificado ou de um número máximo de iteradas (gerações).

Em termos de desempenho, este método é sensível aos parâmetros de controlo a ele associados, tais como, dimensão da população (NP), taxa de cruzamento e fator de escala (F) [Mal+11]. Este algoritmo foi o adotado no âmbito desta dissertação.

CASOS DE ESTUDO

Nesta secção encontram-se apresentados os resultados obtidos através da aplicação do AED (Algoritmo Evolutivo Diferencial), no âmbito da otimização somente dos fatores de escala ou dos fatores de escala juntamente com as funções de pertinência do controlador PID (Proporcional-Integral-Derivativo) difuso, a um sistema MIMO¹ de referência composto por três tanques, AMIRA[®] DTS200. Como forma de comparação recorre-se ao método de otimização analítico de segunda ordem considerado em [Luc12].

6.1 Descrição do sistema

O sistema MIMO de referência AMIRA[®] DTS200, apresentado na figura 6.1, é composto por três tanques cilíndricos de plexiglas, com idêntica secção transversal. Os tanques T_1 e T_2 são alimentados diretamente com água destilada, que se encontra armazenada no reservatório inferior, por meio de duas bombas a eles associadas. Em relação ao tanque central T_3 , este encontra-se ligado aos outros dois tanques através de tubos de secção circular, provido de válvulas de esfera, manualmente ajustáveis. Os níveis de líquido dos três tanques são medidos por transdutores piezoresistivos na gama $[-10; +10]$ V. A saída principal de líquido do sistema encontra-se localizada no tanque T_2 , ligada ao reservatório inferior por meio de um tubo com uma válvula de esfera. Existem três saídas complementares, uma em cada tanque, destinadas à incorporação de possíveis perturbações, sob a forma de um caudal de fuga. Nesta secção pretende-se efetuar o controlo dos níveis de água associados aos tanques T_1 e T_2 , através da leitura dos respetivos sensores e atuando nas bombas correspondentes [Gil+14].

¹MIMO=Multi-Input Multi-Output

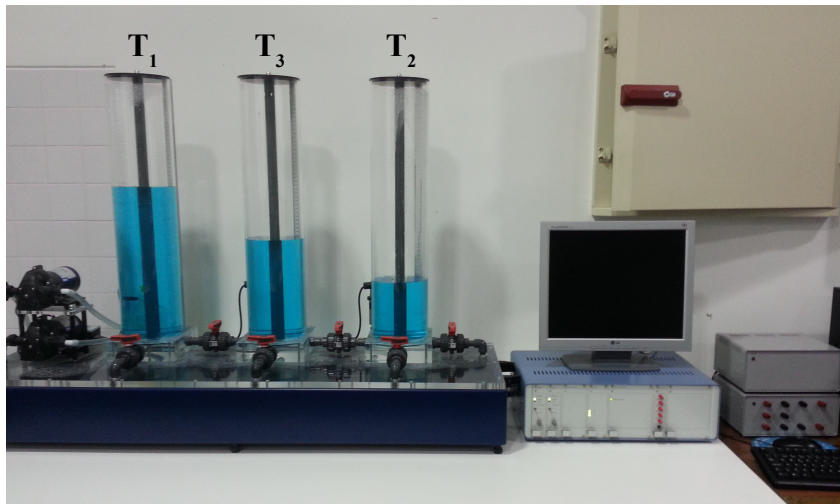


Figura 6.1: Sistema de tanques AMIRA[®] DTS200.

Em relação à ligação entre o servidor (computador) e o sistema, esta foi efetuada através de uma placa de aquisição de dados *National Instruments USB-6009* (Figura 6.2), a qual apresenta uma ligação USB, 8 entradas analógicas de 14 bits, 2 saídas analógicas de 10 bits, 12 ligações digitais I/O e um contador de eventos de 32 bits.



Figura 6.2: *National Instruments USB-6009*.

6.2 Projeto do controlador PID difuso

Como mencionado na secção 3, o controlador PID difuso é composto por duas entradas, nomeadamente o erro e e a variação do erro Δe , e uma saída u^{PID} (ver Figura 3.4). O universo de discurso normalizado das variáveis \tilde{e} , $\Delta \tilde{e}$ e \tilde{u} encontra-se compreendido entre $[-1, 5; +1, 5]$ e segmentado em sete conjuntos difusos (ver Figura 6.3), designadamente $\{NG, NM, NP, ZO, PP, PM, PG\}$. No caso da saída correspondente ao módulo PI, $\Delta \tilde{u}$, foi

decidido que o universo de discurso se encontraria igualmente compreendido no intervalo de $[-1,5; +1,5]$, com uma partição do universo de discurso semelhante à definida anteriormente. As funções de pertinência associadas à saída $\Delta\tilde{u}$ encontram-se representadas na figura 6.4. De sublinhar que as funções de pertinência de ambas as componentes PD e PI são compelidas a ser idênticas, com as larguras apresentadas na tabela 6.1, excetuando a da função de pertinência ZO associada à saída $\Delta\tilde{u}^{PI}$, a qual é fixa, com largura de 0,06.

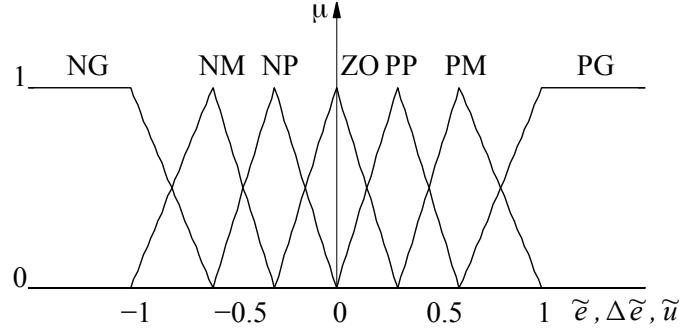


Figura 6.3: Funções de pertinência de \tilde{e} , $\Delta\tilde{e}$ e \tilde{u} .

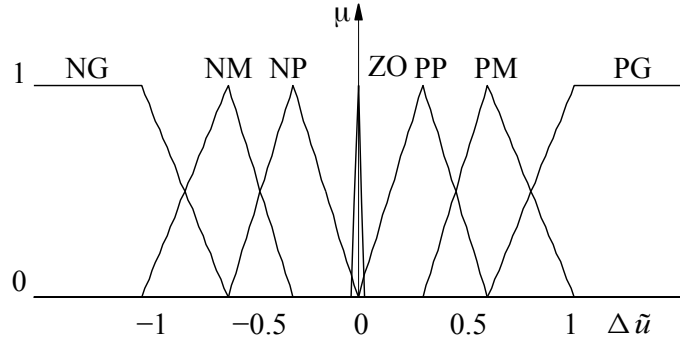


Figura 6.4: Funções de pertinência de $\Delta\tilde{u}$.

Tabela 6.1: Larguras das funções de pertinência.

Termos Linguísticos	NG	NM	NP	ZO	PP	PM	PG
Larguras	0,8	0,6	0,6	0,6	0,6	0,6	0,8

O tipo de mecanismo de inferência utilizado foi o do tipo Mamdani (6.1), enquanto que a saída crespa do controlador é gerada pelo método de desfuzificação centroide.

$$\mu_{e/\Delta e}(e, \Delta e) = \max_k \min(\mu_{A_k}(e), \mu_{B_k}(\Delta e)) \quad (6.1)$$

em que μ_{A_k} e μ_{B_k} representam os valores das funções de pertinência relativos aos conjuntos difusos A e B , respetivamente, associados à regra k .

Em relação às regras do controlador difuso, cada um dos módulos PD e PI tem incorporado quarenta e nove regras, com o formato apresentado na secção 3.1.1 e secção 3.1.2, respetivamente.

A ferramenta do MATLAB *Fuzzy Logic Toolbox* foi utilizada na implementação do controlador PID difuso presente nesta dissertação.

6.3 Otimização em diferido do controlador PID

O modelo não linear do sistema de três tanques é descrito por um modelo NNARX, composto por três camadas, nomeadamente, uma camada de entrada, uma camada interna e uma camada de saída. Por forma a capturar a dinâmica do sistema, a rede neuronal foi treinada em diferido utilizando o algoritmo de Levenberg-Marquardt, através da ferramenta disponível no MATLAB *Neural Network Toolbox*. O preditor da rede neuronal usado nos testes representado por:

$$y_{net}(k) = A \cdot \tanh \left[B \cdot U_{net}(k) + c \right] \quad (6.2)$$

$$\text{com,} \quad A = \begin{bmatrix} 6.24 & -2.45 \\ 2.87 & -1.90 \end{bmatrix} \quad B = \begin{bmatrix} 0.327 & 0.540 \\ 0.059 & 0.036 \\ -0.625 & -1.466 \\ 0.132 & 0.217 \\ 0.001 & -0.002 \\ 0.004 & 0.008 \\ -0.002 & -0.002 \\ -0.001 & -0.005 \end{bmatrix}^T \quad c = \begin{bmatrix} -0.511 \\ -0.435 \end{bmatrix} \times 10^{-3}$$

6.3.1 PID difuso com sintonização dos fatores de escala

A otimização do controlador PID difuso é obtida através da resolução de um problema de otimização não linear com restrições, sendo a função de custo definida por um índice de desempenho quadrático, que inclui penalizações tanto nos erros de controlo como nos incrementos das ações de controlo. O problema de otimização com restrições pode ser formulado do seguinte modo:

$$\min_K J = \min_{(K)} \left\{ \sum_{k=0}^{N_p} 20 \left[y_1(k) - r_1(k) \right]^2 + 20 \left[y_2(k) - r_2(k) \right]^2 + \right. \\ \left. + 0.1 \left[u_1(k) - u_1(k-1) \right]^2 + 0.1 \left[u_2(k) - u_2(k-1) \right]^2 \right\} \quad (6.3)$$

com os fatores de escala $K = [K_{e1}, K_{\Delta e1}, K_{u1}, K_{\Delta u1}, K_{e2}, K_{\Delta e2}, K_{u2}, K_{\Delta u2}]$, sujeito à dinâmica do sistema e às condições de restrição,

$$\begin{aligned} 0 &\leq y(k) \leq 1, k = 1, \dots, N_p \\ 0 &\leq u(k) \leq 5, k = 0, \dots, N_p - 1 \\ K_{e1}, K_{e2} &\geq 0 \\ K_{\Delta e1}, K_{\Delta e2} &\geq 0 \\ K_{u1}, K_{u2} &\geq 0 \\ K_{\Delta u1}, K_{\Delta u2} &\geq 0 \end{aligned} \quad (6.4)$$

sendo y o vetor de saída, r o sinal de referência e N_p o horizonte de predição. A ação de controlo é dada por:

$$u(k) = u(k-1) + K_{\Delta u} \cdot \Delta \tilde{u}(k) + K_u \cdot \tilde{u}(k) \quad (6.5)$$

Sintonização usando o AED

Dada uma população inicial de vetores de fatores de escala, o algoritmo de otimização com restrições calcula, em cada iteração, um novo e melhorado conjunto de fatores de escala, até que um mínimo do critério escolhido seja alcançado, dentro de um tempo limite pré-especificado, ou até que um número máximo de iterações seja atingido.

Os resultados obtidos a partir de experiências conduzidas no sistema de três tanques AMIRA[®] DTS200, relativas à utilização do AED, encontram-se ilustrados na figura 6.5 e figura 6.6. Os valores ótimos resultantes desta sintonização encontram-se apresentados na tabela 6.2.

Tabela 6.2: Fatores de escala ótimos com sintonização em diferido usando o AED.

Fatores de escala	Tanque 1				Tanque 2			
	Ke	KΔe	KΔu	Ku	Ke	KΔe	KΔu	Ku
Valores ótimos	0,1789	1,5154	2,0478	8,0000	0,8318	6,0142	0,5907	8,0000

Sintonização usando o método analítico de segunda ordem

Dado um vetor inicial de fatores de escala, o algoritmo de otimização com restrições calcula, em cada iteração, um conjunto melhorado de fatores de escala até que um mínimo local do índice de desempenho escolhido seja alcançado, tendo em conta um limite máximo de iterações pré-especificado.

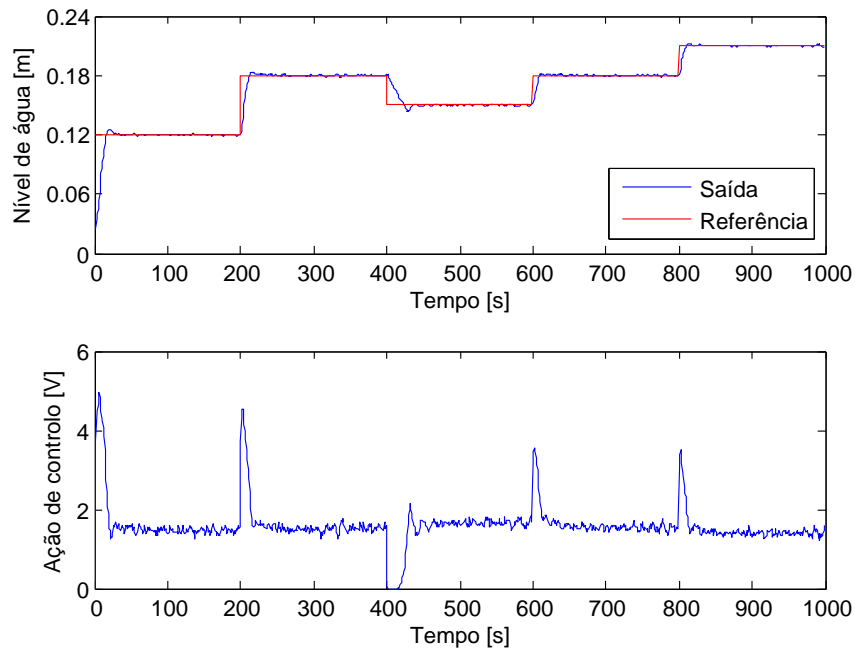


Figura 6.5: Tanque 1 do sistema MIMO controlado por um controlador PID difuso com sintonização dos fatores de escala em diferido, considerando o AED.

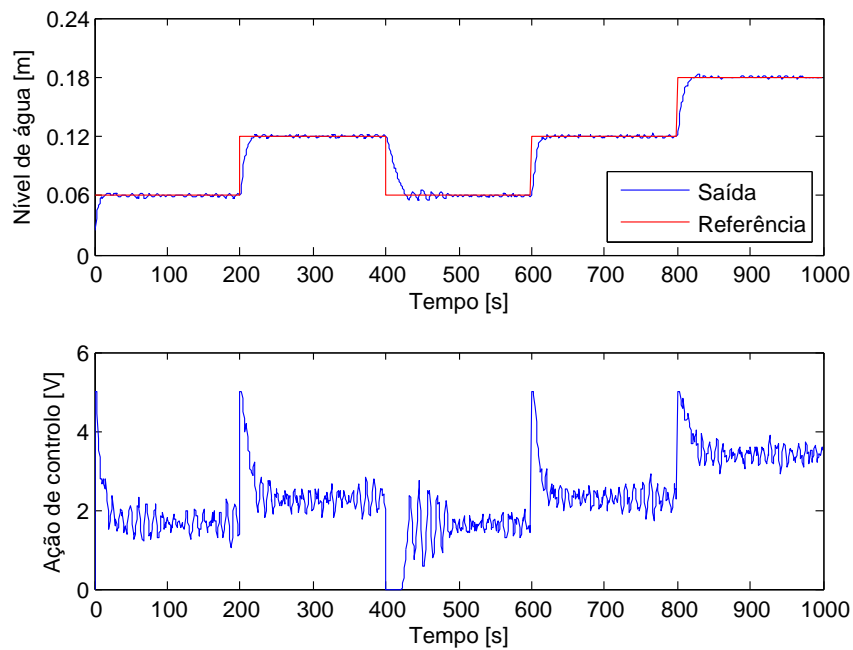


Figura 6.6: Tanque 2 do sistema MIMO controlado por um controlador PID difuso com sintonização dos fatores de escala em diferido, considerando o AED.

Os resultados das simulações, obtidos através das experiências realizadas no sistemas de três tanques, encontram-se representados na figura 6.7 e figura 6.8. Os valores ótimos resultantes desta sintonização encontra-se apresentados na tabela 6.3.

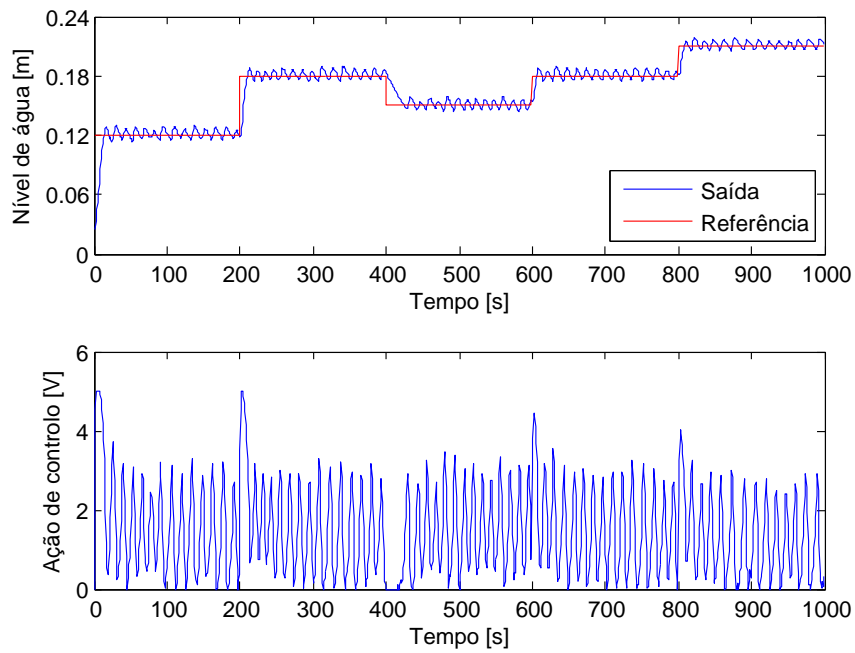


Figura 6.7: Tanque 1 do sistema MIMO controlado por um controlador PID difuso com sintonização dos fatores de escala em diferido, considerando o método analítico de segunda ordem.

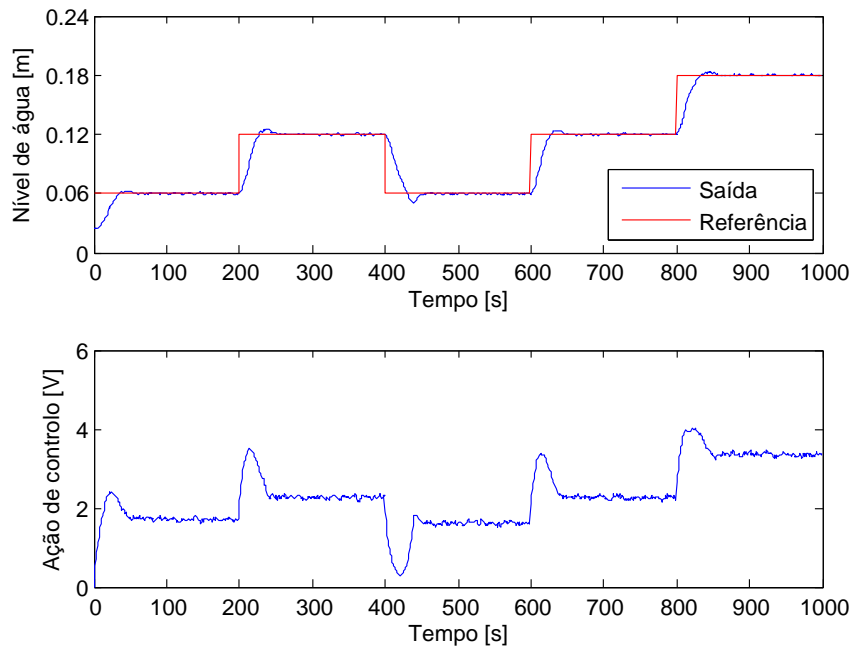


Figura 6.8: Tanque 2 do sistema MIMO controlado por um controlador PID difuso com sintonização dos fatores de escala em diferido, considerando o método analítico de segunda ordem.

Tabela 6.3: Fatores de escala ótimos com sintonização em diferido usando o método analítico de segunda ordem.

Fatores de escala	Tanque 1				Tanque 2			
	Ke	K Δe	K Δu	Ku	Ke	K Δe	K Δu	Ku
Valores ótimos	2,0200	13,0963	1,6051	1,4423	0,2368	2,0858	1,0403	0,4122

6.3.2 PID difuso com sintonização dos fatores de escala e funções de pertinência em diferido

O problema de otimização com restrições foi formulado considerando uma função de custo sob a forma de uma métrica quadrada, que inclui o vetor de erro de controlo e o vetor do incremento da ação de controlo. O problema de otimização com restrições pode então ser definido por:

$$\min_{(K,L)} J = \min_{(K,L)} \left\{ \sum_{k=0}^{N_p} 20 \left[y_1(k) - r_1(k) \right]^2 + 20 \left[y_2(k) - r_2(k) \right]^2 + 0.1 \left[u_1(k) - u_1(k-1) \right]^2 + 0.1 \left[u_2(k) - u_2(k-1) \right]^2 \right\} \quad (6.6)$$

Os fatores de escala K e as larguras das funções de pertinência L são representados por $K = [K_{e1}, K_{\Delta e1}, K_{u1}, K_{\Delta u1}, K_{e2}, K_{\Delta e2}, K_{u2}, K_{\Delta u2}]$ e $L = [L_{NG}, L_{NM}, L_{NP}, L_{ZO}, L_{PP}, L_{PM}, L_{PG}]$, devendo obedecer às seguintes restrições:

$$\begin{aligned} 0 &\leq y(k) \leq 1, k = 1, \dots, N_p \\ 0 &\leq u(k) \leq 5, k = 0, \dots, N_p - 1 \\ K_{e1}, K_{e2} &\geq 0 \\ K_{\Delta e1}, K_{\Delta e2} &\geq 0 \\ K_{u1}, K_{u2} &\geq 0 \\ K_{\Delta u1}, K_{\Delta u2} &\geq 0 \\ 0.25 &\leq \mu_{(NG=NM)} \leq 0.5 \\ 0.25 &\leq \mu_{(NM=NP)} \leq 0.5 \\ 0.25 &\leq \mu_{(NP=ZO)} \leq 0.5 \\ 0.25 &\leq \mu_{(ZO=PP)} \leq 0.5 \\ 0.25 &\leq \mu_{(PP=PM)} \leq 0.5 \\ 0.25 &\leq \mu_{(PM=PG)} \leq 0.5 \end{aligned} \quad (6.7)$$

onde $(NG = NM), \dots, (PM = PG)$ representa a interseção entre as funções de pertinência correspondentes e a ação de controlo é dada por:

$$u(k) = u(k-1) + K_{\Delta u} \cdot \Delta \tilde{u}(k) + K_u \cdot \tilde{u}(k) \quad (6.8)$$

Sintonização usando o AED

Neste caso, tanto os fatores de escala como as larguras das funções de pertinência do controlador PID difuso são sintonizados, através da inicialização de uma população de vetores que inclui estimativas iniciais destes parâmetros. A cada iteração é obtido um novo conjunto melhorado de parâmetros, até que um mínimo da função de custo seja alcançado, dentro de um tempo limite pré-especificado, ou até que um número máximo de iterações seja atingido. No fim de cada iteração são atualizadas ambas as componentes difusas PD e PI.

Os resultados obtidos através desta sintonização encontram-se representados na figura 6.9 e figura 6.10. Os valores que resultaram da sintonização dos parâmetros do controlador PID, fatores de escala e larguras das funções de pertinência, encontram-se apresentados nas tabelas 6.4 e 6.5, respetivamente.

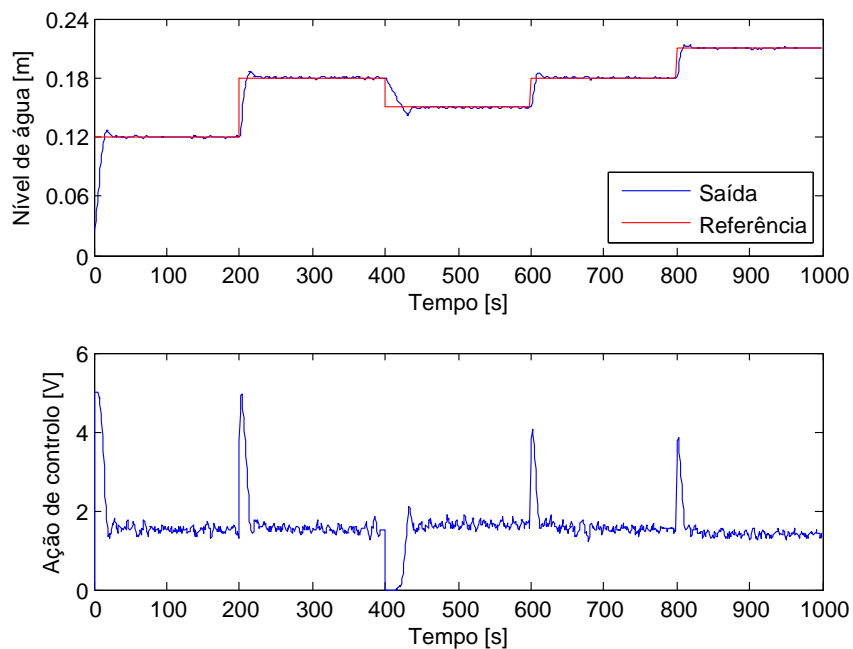


Figura 6.9: Tanque 1 do sistema MIMO controlado por um controlador PID difuso com sintonização dos fatores de escala e das funções de pertinência em diferido, considerando o AED.

Tabela 6.4: Fatores de escala ótimos com sintonização em diferido dos fatores de escala e funções de pertinência usando o AED.

Fatores de escala	Tanque 1				Tanque 2			
	K_e	$K\Delta e$	$K\Delta u$	K_u	K_e	$K\Delta e$	$K\Delta u$	K_u
Valores ótimos	0,3867	2,7089	0,7307	6,4193	0,7366	4,9701	0,1018	7,3546

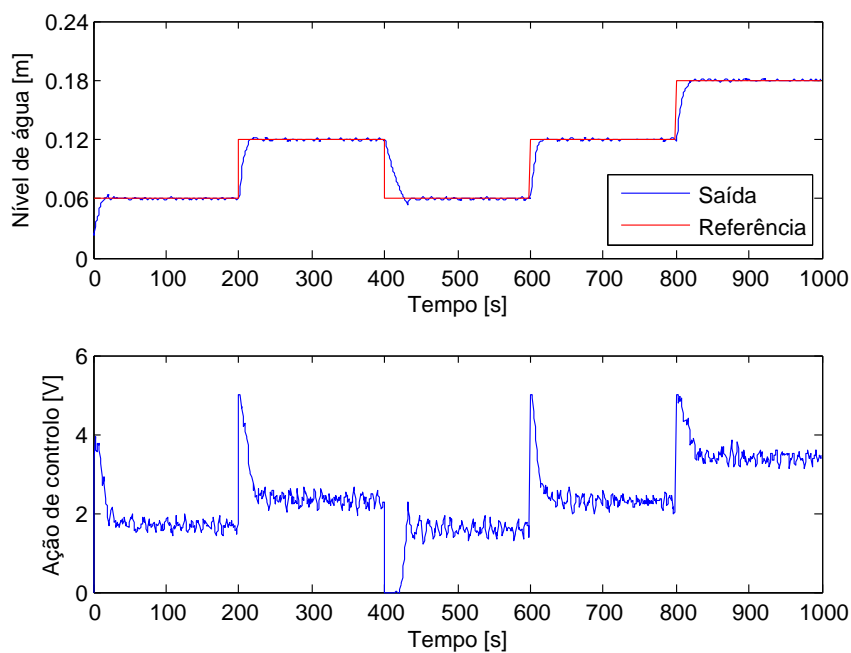


Figura 6.10: Tanque 2 do sistema MIMO controlado por um controlador PID difuso com sintonização dos fatores de escala e das funções de pertinência em diferido, considerando o AED.

Tabela 6.5: Larguras ótimas das funções de pertinência com sintonização em diferido usando o AED.

Termos Linguísticos	NG	NM	NP	ZO	PP	PM	PG
Larguras	0,5350	0,5350	0,6	0,4	0,6	0,5350	0,5670

As funções de pertinência sintonizadas, cujas larguras se apresentam na tabela 6.5, associadas ao controlador PID difuso, encontram-se ilustradas nas figuras 6.11 e 6.12.

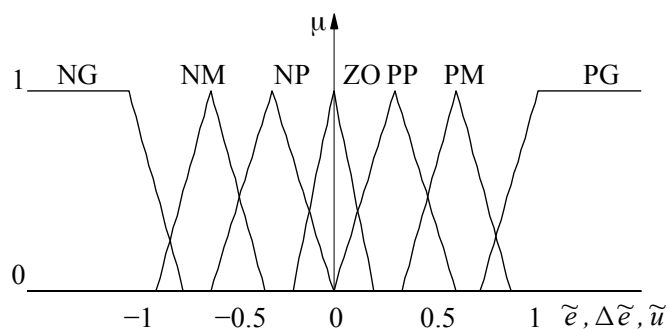
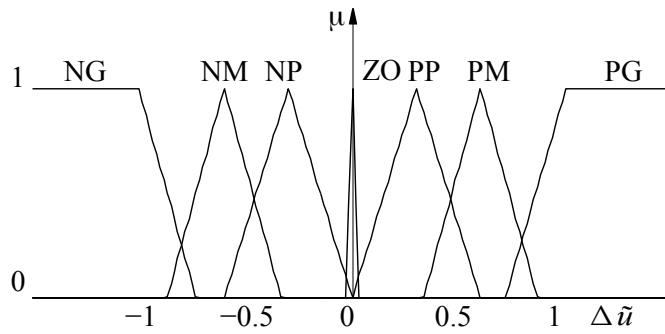


Figura 6.11: Funções de pertinência de \tilde{e} , $\Delta\tilde{e}$ e \tilde{u} .

Figura 6.12: Funções de pertinência de $\Delta \tilde{u}$.

Sintonização usando o método analítico de segunda ordem

Nesta metodologia o vetor de parâmetros inicial incluirá, além dos fatores de escala do controlador PID difuso, as larguras das funções de pertinência. Para além disso, como a sintonização das funções de pertinência implica o rearranjo dos conjuntos difusos subjacentes, estas devem ser atualizadas no fim de cada iteração.

Os resultados obtidos através desta sintonização encontram-se apresentados na figura 6.13 e figura 6.14. Os valores que resultaram da sintonização dos parâmetros do controlador PID encontram-se apresentados nas tabelas 6.6 e 6.7, respetivamente.

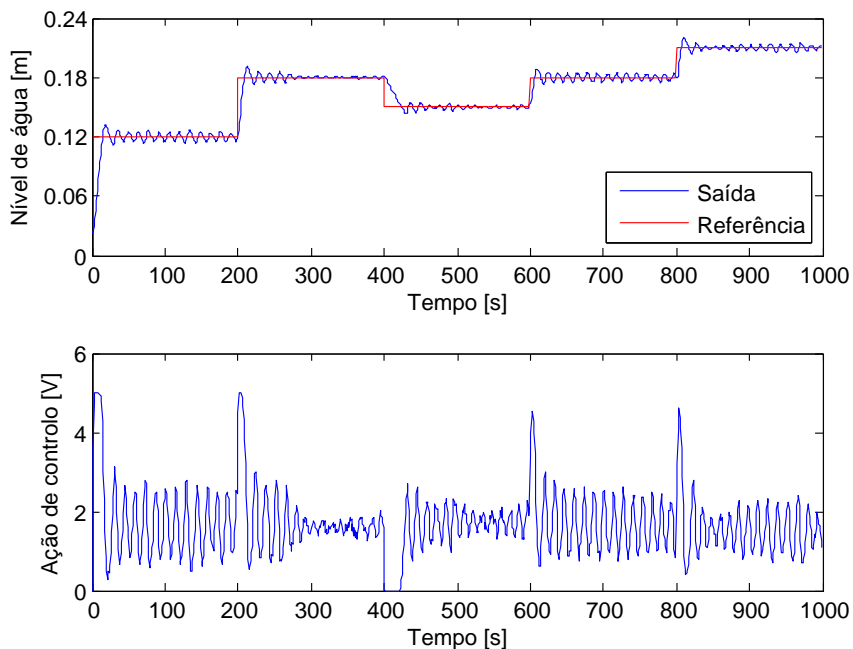


Figura 6.13: Tanque 1 do sistema MIMO controlado por um controlador PID difuso com sintonização dos fatores de escala e das funções de pertinência em diferido, considerando o método analítico de segunda ordem.

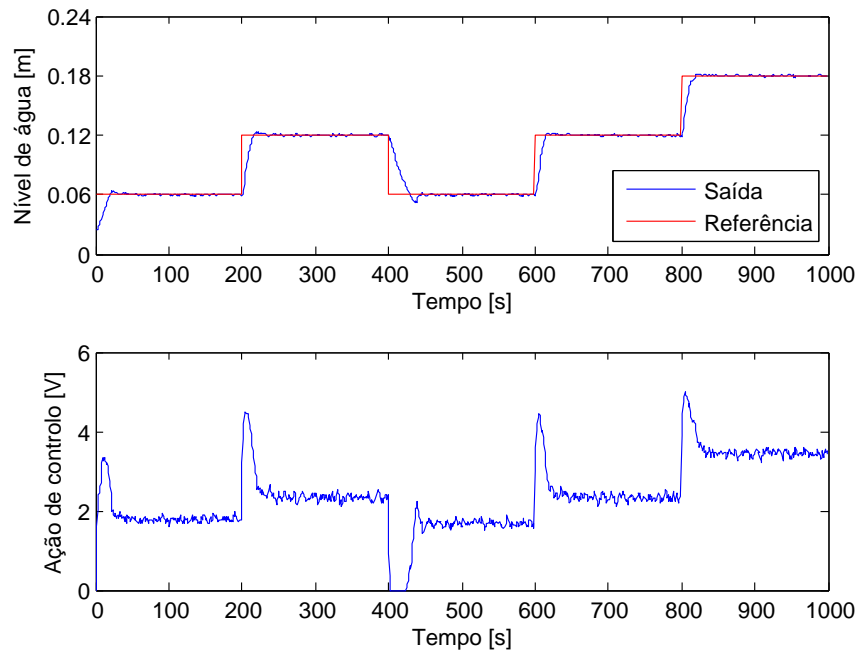


Figura 6.14: Tanque 2 do sistema MIMO controlado por um controlador PID difuso com sintonização dos fatores de escala e das funções de pertinência em diferido, considerando o método analítico de segunda ordem.

Tabela 6.6: Fatores de escala ótimos com sintonização em diferido dos fatores de escala e funções de pertinência usando o método analítico de segunda ordem.

Fatores de escala	Tanque 1				Tanque 2			
	K_e	$K\Delta e$	$K\Delta u$	K_u	K_e	$K\Delta e$	$K\Delta u$	K_u
Valores ótimos	0,4833	1,7878	3,0000	3,0000	0,4100	2,8266	1,2060	3,0000

Tabela 6.7: Larguras ótimas das funções de pertinência com sintonização em diferido usando o método analítico de segunda ordem.

Termos Linguísticos	NG	NM	NP	ZO	PP	PM	PG
Larguras	0,5851	0,5832	0,5998	0,4	0,5998	0,5359	0,5728

O resultado da sintonização das funções de pertinência, cujas larguras se encontram na tabela 6.7, associadas ao controlador PID difuso, são apresentadas nas figuras 6.15 e 6.16.

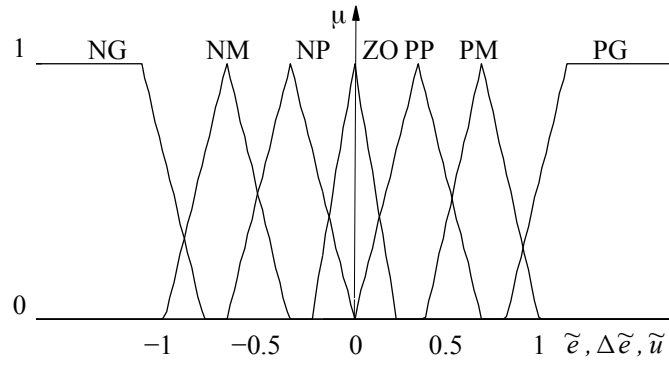


Figura 6.15: Funções de pertinência de \tilde{e} , $\Delta\tilde{e}$ e \tilde{u} .

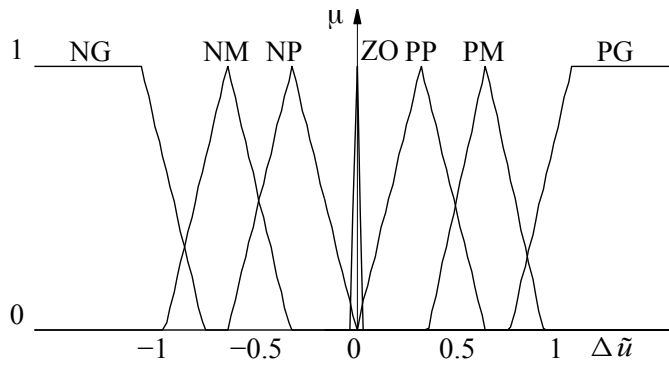


Figura 6.16: Funções de pertinência de $\Delta\tilde{u}$.

6.4 Otimização em-linha do controlador PID difuso usando o método analítico de segunda ordem

O problema de otimização com restrições é computacionalmente exigente em termos temporais, podendo acontecer que, quando implementado em-linha, para períodos de amostragem suaves ou rígidos, o intervalo de amostragem definido pelo utilizador seja excedido pela rotina de otimização. Por esta razão utiliza-se uma arquitetura baseada no paradigma de computação distribuída, em que a implementação do controlador difuso e o ciclo de otimização se encontram localizados em computadores diferentes, sendo a dinâmica do sistema aproximada por um preditor linear, nomeadamente um modelo ARX, para cada tanque. De sublinhar que este modelo é treinado em-linha usando o método dos mínimos quadráticos recursivos apresentado na secção 4.1.1.

Em relação à arquitetura, a otimização em-linha é composta por duas rotinas de otimização, nomeadamente, a rotina principal e a rotina de otimização. No que diz respeito à rotina de otimização, esta é executada no computador denominado por "cliente", enquanto que a rotina de otimização, que visa sintonizar os parâmetros do controlador PID difuso, é executada no computador "servidor". Relativamente à comunicação entre os computadores

cliente e servidor, foi adotada uma rede TCP/IP² ponto-a-ponto (ver [Gil+14]).

6.4.1 PID difuso com sintonização dos fatores de escala e funções de pertinência em-linha

O índice de desempenho objeto de minimização foi escolhido como uma função quadrática, incluindo penalizações tanto nos erros de controlo como nos incrementos das ações de controlo. A formulação do problema de otimização pode então ser escrita como:

$$\min_{(K_1, L_1)} \left\{ \sum_{k=0}^{N_p} 20 \left[y_1(k) - r_1(k) \right]^2 + 0.1 \left[u_1(k) - u_1(k-1) \right]^2 \right\} \quad (6.9)$$

$$\min_{(K_2, L_2)} \left\{ \sum_{k=0}^{N_p} 30 \left[y_2(k) - r_2(k) \right]^2 + 0.1 \left[u_2(k) - u_2(k-1) \right]^2 \right\} \quad (6.10)$$

em que (6.9) e (6.10) são referentes aos tanques T_1 e T_2 , respetivamente. No que diz respeito aos fatores de escala K_1 e K_2 , e às larguras das funções de pertinência L_1 e L_2 , estas são representadas por $K_1 = [K_{e1}, K_{\Delta e1}, K_{u1}, K_{\Delta u1}]$, $K_2 = [K_{e2}, K_{\Delta e2}, K_{u2}, K_{\Delta u2}]$ e $L_{1,2} = [L_{NG}, L_{NM}, L_{NP}, L_{ZO}, L_{PP}, L_{PM}, L_{PG}]$, devendo ser cumpridas as seguintes restrições:

$$\begin{aligned} 0 &\leq y(k) \leq 1, k = 1, \dots, N_p \\ 0 &\leq u(k) \leq 5, k = 0, \dots, N_p - 1 \\ K_{e1}, K_{e2} &\geq 0 \\ K_{\Delta e1}, K_{\Delta e2} &\geq 0 \\ K_{u1}, K_{u2} &\geq 0 \\ K_{\Delta u1}, K_{\Delta u2} &\geq 0 \\ 0.25 &\leq \mu_{(NG=NM)} \leq 0.5 \\ 0.25 &\leq \mu_{(NM=NP)} \leq 0.5 \\ 0.25 &\leq \mu_{(NP=ZO)} \leq 0.5 \\ 0.25 &\leq \mu_{(ZO=PP)} \leq 0.5 \\ 0.25 &\leq \mu_{(PP=PM)} \leq 0.5 \\ 0.25 &\leq \mu_{(PM=PG)} \leq 0.5 \end{aligned} \quad (6.11)$$

onde $(NG = NM), \dots, (PM = PG)$ representa a interseção entre as funções de pertinência correspondentes e a ação de controlo é dada por:

$$u(k) = u(k-1) + K_{\Delta u} \cdot \Delta \tilde{u}(k) + K_u \cdot \tilde{u}(k) \quad (6.12)$$

Como já foi referido, existem duas rotinas a funcionar separadamente, uma relativa ao ciclo de controlo (rotina principal) e a outra subordinada ao ciclo de otimização (rotina de otimização). Em relação à rotina principal, esta começa por gerar um endereço de Internet

²TCP/IP=Transmission Control Protocol/Internet Protocol

socket no computador cliente, o qual é necessário para permitir o envio de pacotes para o servidor, assim como receber pacotes do mesmo. Em seguida, os parâmetros iniciais do controlador e do modelo são enviados para o servidor, por forma a permitir a execução da rotina de otimização antes de entrar no ciclo de controlo. A representação da rotina principal encontra-se ilustrada no Algoritmo 3.

Algorithm 3 Rotina principal do cliente.

```

Require:  $K, L$  % Fatores de escala e funções de pertença
 $csock \leftarrow \text{novo\_socket}(\text{"servidor\_IP"}, \text{número\_porta})$ 
Inicializa fatores de escala e funções de pertença:  $K_e, K_{\Delta e}, K_{\Delta u}, K_u,$ 
 $L_{NG}, L_{NM}, L_{NP}, L_{ZO}, L_{PP}, L_{PM}, L_{PG}$ 
Inicializa parâmetros do modelo:  $\theta$ 
 $\text{imprime\_escrita.csock}(\theta, K, L)$ 
 $k \leftarrow 0$ 
repeat
   $y(k) \leftarrow \text{lerdaq}(\text{canal})$  % Ler DAQ
   $\theta \leftarrow \text{atualização\_modelo}()$  % Atualização recursiva do modelo
  if (novo  $K$  e  $L$  disponível) then
     $[K, L] \leftarrow \text{buffer\_leitura.csock}(\text{buffer})$  % Novos fatores de escala e funções de pertença
     $\text{imprime\_escrita.csock}(\theta, K, L)$ 
  end if
   $u(k) \leftarrow \text{PIDdifuso}(K, L, e(k), \Delta e(k), u(k-1))$  % Ação de controlo atual
   $\text{escreverdaq}(\text{canal}, u(k)) \leftarrow u(k)$  % Escrever DAQ
   $\text{pausa}(\text{tempo\_amostragem} - \text{tempo\_simulação})$ 
until  $\text{Fim\_Controlo}$ 
 $csock.\text{fechar}()$ 

```

No que diz respeito à rotina de otimização, esta começa de forma similar à rotina principal, ou seja, é criado um endereço de Internet *socket* no servidor, o qual é usado para trocar pacotes de dados com o cliente. O ciclo de otimização recebe os parâmetros do controlador e do modelo a partir do *buffer* do *socket*, seguindo-se a execução de uma rotina de minimização, a qual fornece os parâmetros do controlador PID difuso otimizados. Estes parâmetros serão posteriormente enviados para o cliente. Esta rotina de minimização foi realizada usando duas abordagens distintas, uma em que os parâmetros do controlador PID difuso são sintonizados simultaneamente, e outra em que os parâmetros são sintonizados através de uma topologia em cascata. A representação da rotina de otimização encontra-se ilustrada no Algoritmo 4.

Sintonização dos fatores de escala e das funções de pertença em simultâneo

Esta abordagem otimiza simultaneamente os parâmetros do controlador PID difuso. Os resultados obtidos através desta otimização encontram-se apresentados nas figuras 6.17 e 6.18.

Algorithm 4 Rotina de otimização.

```
Ensure:  $[K, L] \Leftarrow \arg \min J(K, L, \theta)$ 
Require:  $K, L, \theta$ 
 $ssock \Leftarrow \text{ novo\_socket}(\text{número\_porta})$ 
 $itermax \Leftarrow \text{ValorMax}$ 
repeat
   $iter \Leftarrow 1$ 
   $(\theta, K, L) \Leftarrow \text{buffer\_leitura.ssock}(\text{buffer})$ 
  while  $iter \leq itermax$  do
     $J \Leftarrow 0$ 
    for  $k = 1 : N_p$  do
       $e(k) \Leftarrow r(k) - y(k)$ 
       $\Delta e(k) \Leftarrow e(k) - e(k-1)$ 
       $u(k) \Leftarrow \text{PIDdifuso}(K, L, e(k), \Delta e(k), u(k-1))$ 
       $fi \Leftarrow [y(k - n_a + 1 : k) \ u(k - n_b + 1 : k)]$ 
       $y(k+1) \Leftarrow g(fi, \theta)$ 
       $J \Leftarrow J + \mathcal{L}_k(e(k), \Delta u(k), u(k))$ 
    end for
    case:
      Sintonização simultânea:
       $[K, L] \Leftarrow fmin(J)$ 
      Sintonização sequencial:
       $K \Leftarrow fmin(J)$ , com  $L$  constante
       $L \Leftarrow fmin(J)$ , com  $K$  constante
    end case
     $iter \Leftarrow iter + 1$ 
  end while
   $\text{imprime\_escrita.ssock}(K, L)$ 
until  $Fim\_Otimização$ 
 $ssock.\text{fechar}()$ 
```

A representação da evolução dos fatores de escala encontra-se apresentada nas figuras 6.19 e 6.20, enquanto que a evolução relativa às larguras das funções de pertinência se encontra representada nas figuras 6.21 e 6.22.

Sintonização dos fatores de escala e das funções de pertinência considerando uma configuração em cascata

Neste caso a sintonização dos parâmetros do controlador PID difuso é realizada sequencialmente, ou seja, primeiro procede-se à otimização dos fatores de escala, mantendo as funções de pertinência fixas, e em seguida sintoniza-se as larguras das funções de pertinência, assumindo os fatores de escala inalterados com os valores anteriormente sintonizados. Os resultados obtidos através desta otimização encontram-se apresentados nas figuras 6.23 e 6.24.

6.4. OTIMIZAÇÃO EM-LINHA DO CONTROLADOR PID DIFUSO USANDO O MÉTODO ANALÍTICO DE SEGUNDA ORDEM

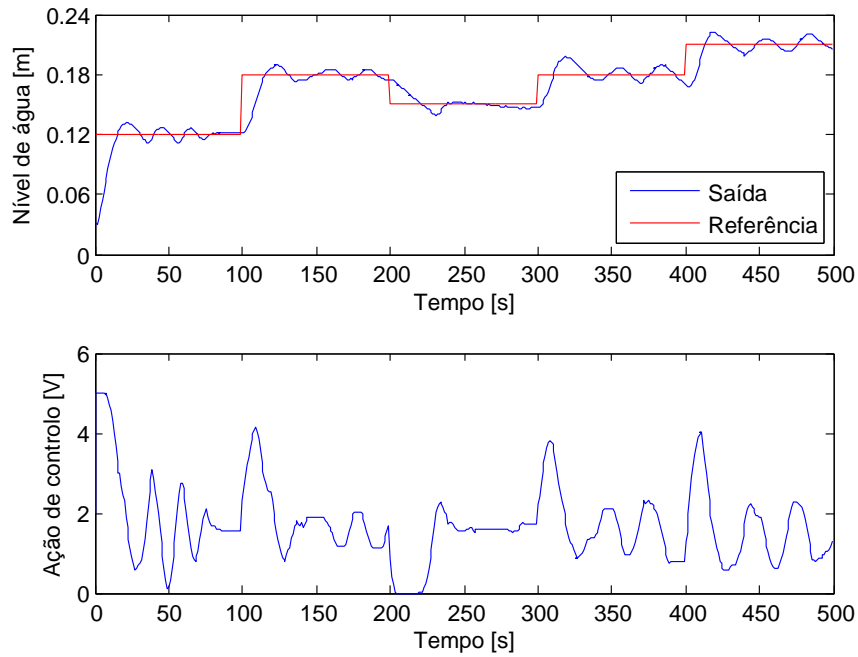


Figura 6.17: Tanque 1 do sistema MIMO controlado por um controlador PID difuso com sintonização dos fatores de escala e das funções de pertinência em-linha.

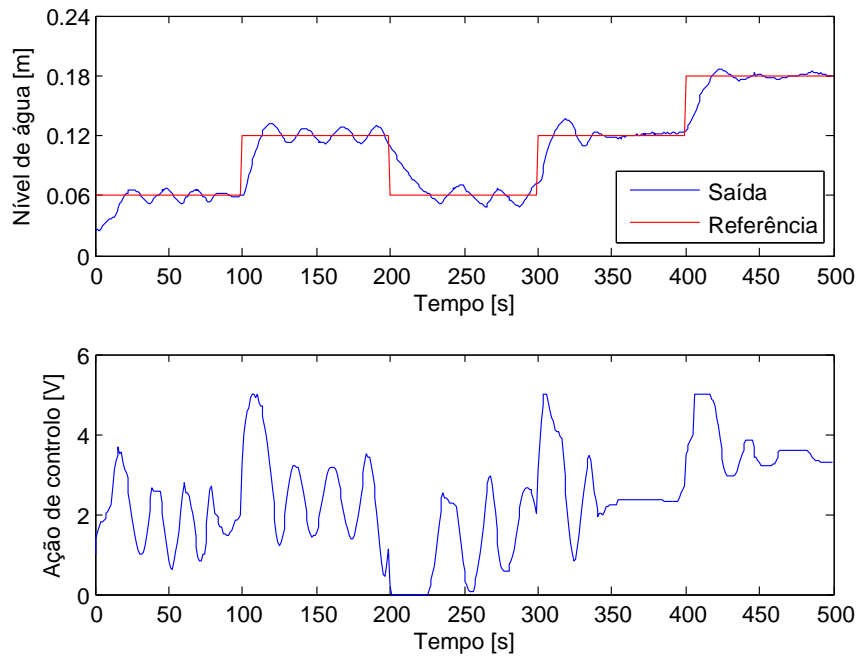


Figura 6.18: Tanque 2 do sistema MIMO controlado por um controlador PID difuso com sintonização dos fatores de escala e das funções de pertinência em-linha.

A representação da evolução dos fatores de escala encontra-se apresentada nas figuras 6.25 e 6.26, enquanto que a evolução relativa às larguras das funções de pertinência se encontra representada nas figuras 6.27 e 6.28.

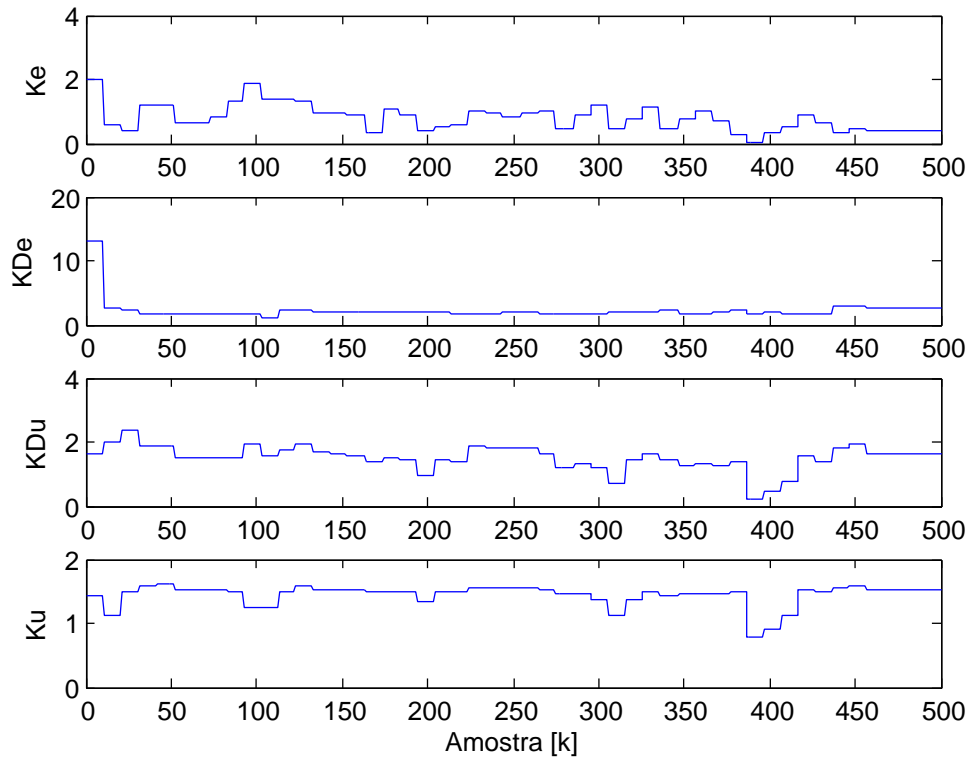


Figura 6.19: Evolução dos fatores de escala associados ao Tanque 1.

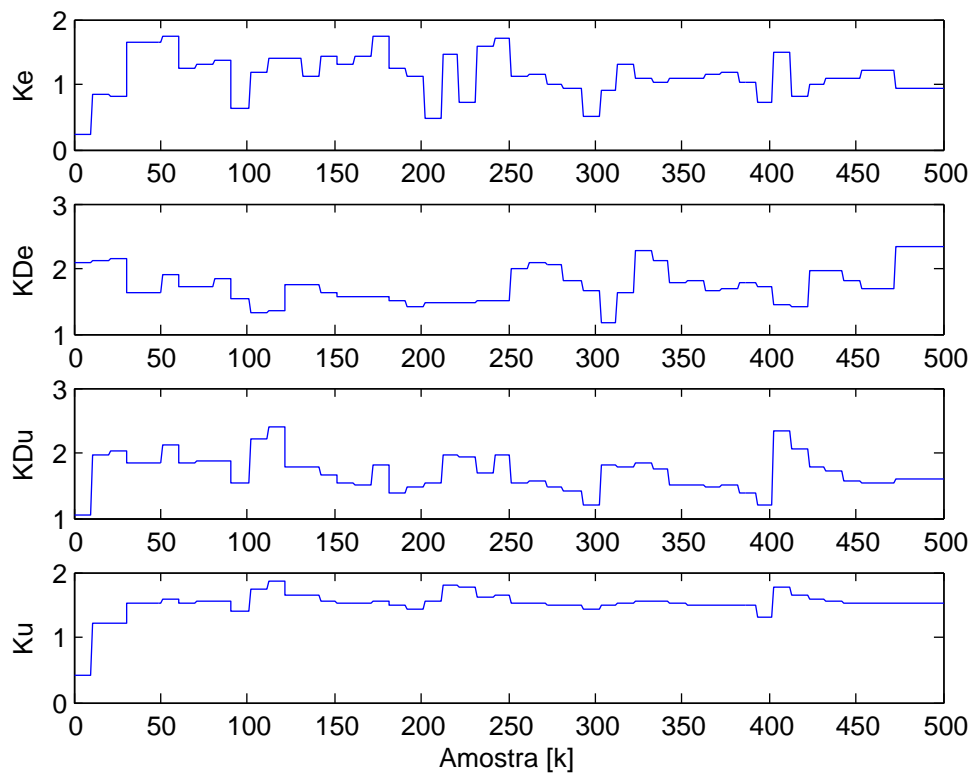


Figura 6.20: Evolução dos fatores de escala associados ao Tanque 2.

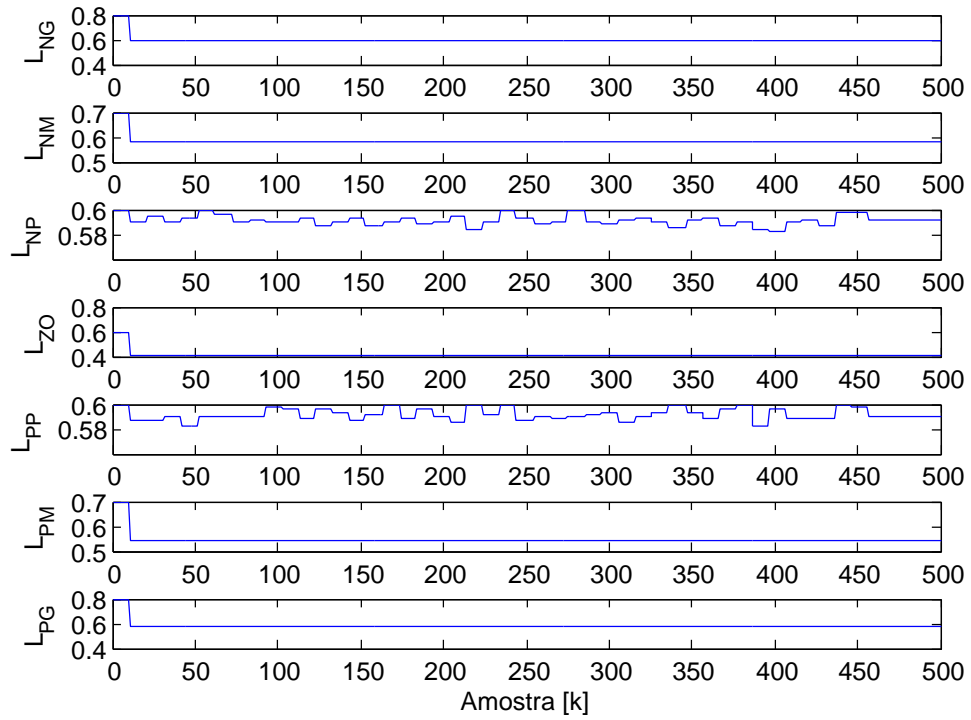


Figura 6.21: Evolução das larguras das funções de pertinência associadas ao Tanque 1.

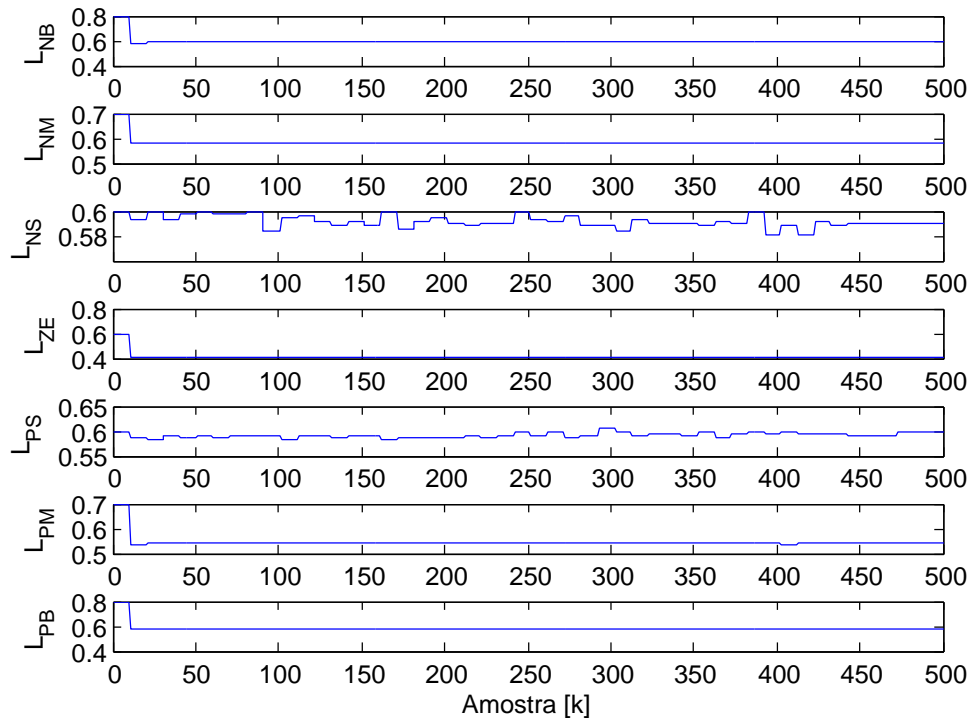


Figura 6.22: Evolução das larguras das funções de pertinência associadas ao Tanque 2.

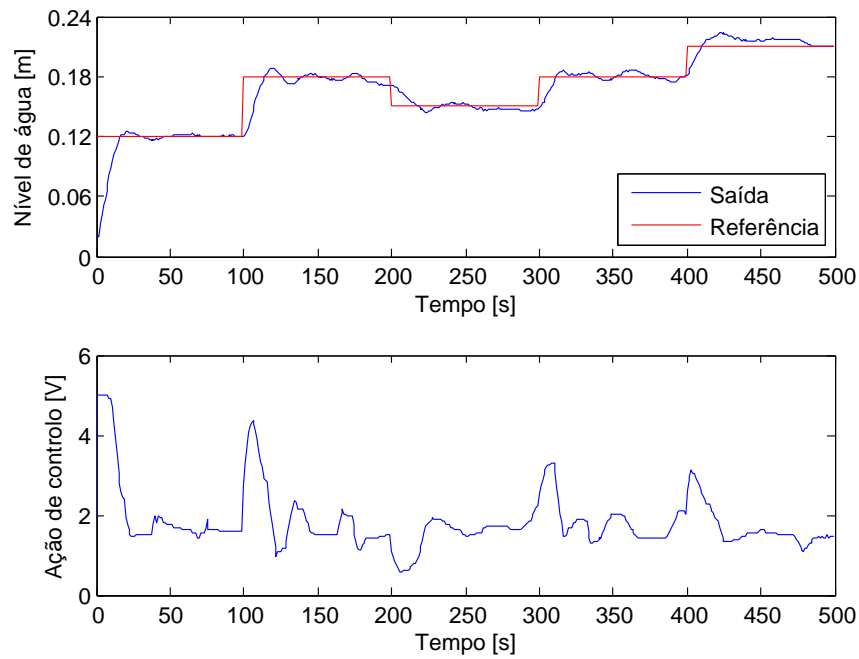


Figura 6.23: Tanque 1 do sistema MIMO controlado por um controlador PID difuso com sintonização dos fatores de escala e das funções de pertinência em-linha, considerando uma configuração em cascata.

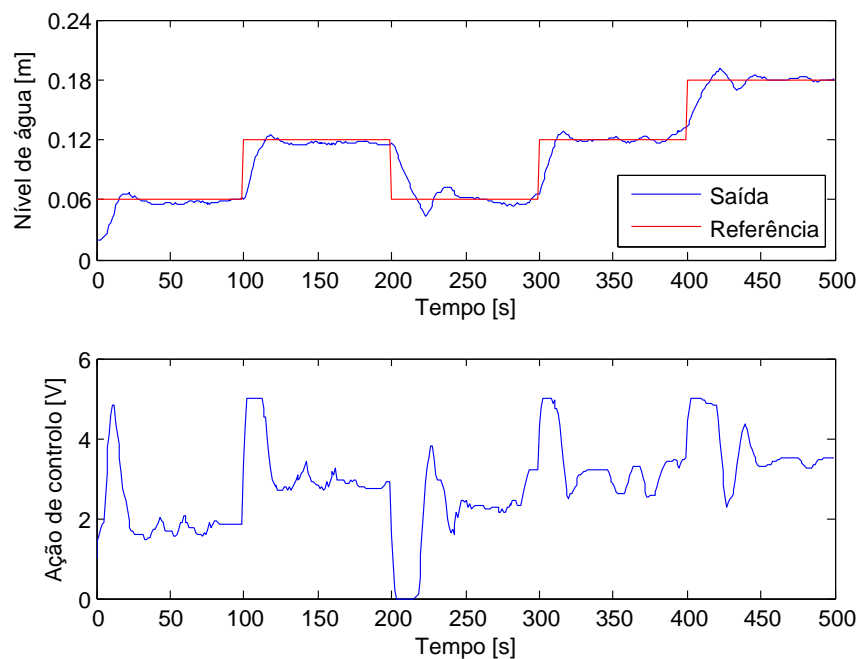


Figura 6.24: Tanque 2 do sistema MIMO controlado por um controlador PID difuso com sintonização dos fatores de escala e das funções de pertinência em-linha, considerando uma configuração em cascata.

6.4. OTIMIZAÇÃO EM-LINHA DO CONTROLADOR PID DIFUSO USANDO O MÉTODO ANALÍTICO DE SEGUNDA ORDEM

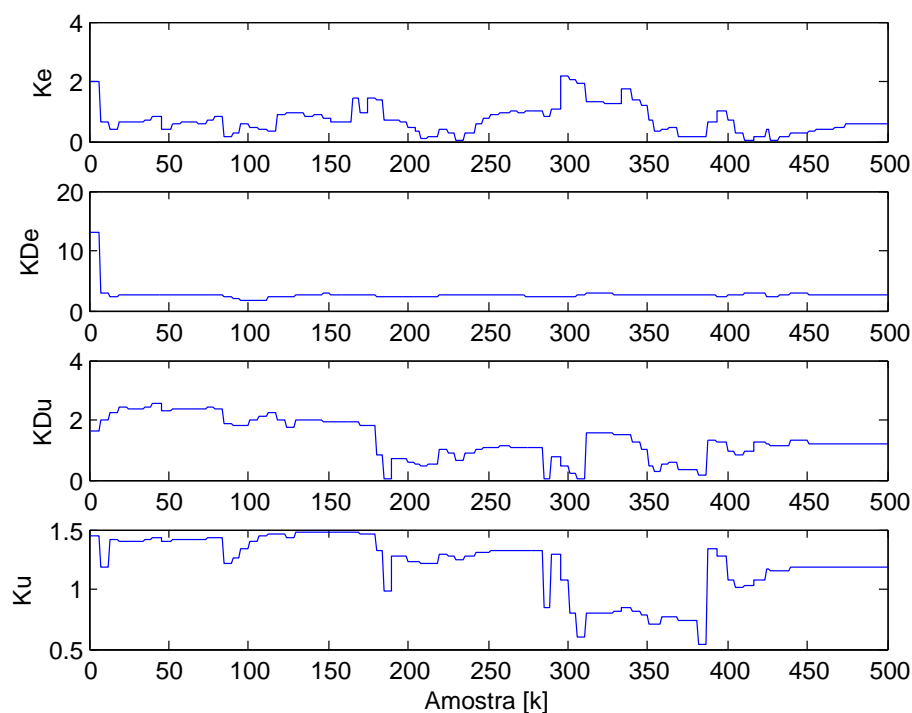


Figura 6.25: Evolução dos fatores de escala associados ao Tanque 1, considerando uma configuração em cascata.

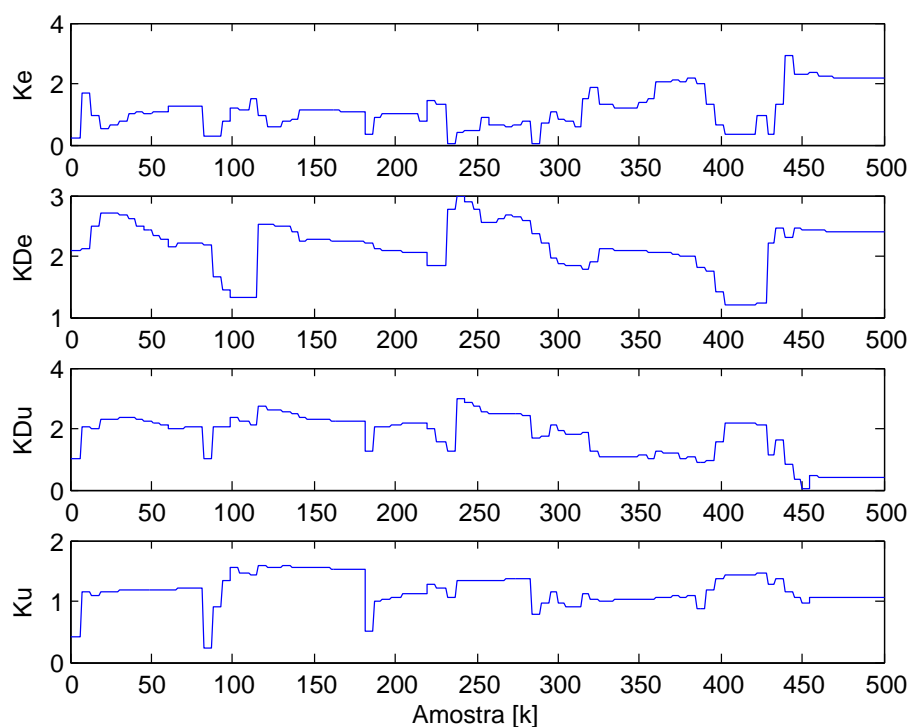


Figura 6.26: Evolução dos fatores de escala associados ao Tanque 2, considerando uma configuração em cascata.

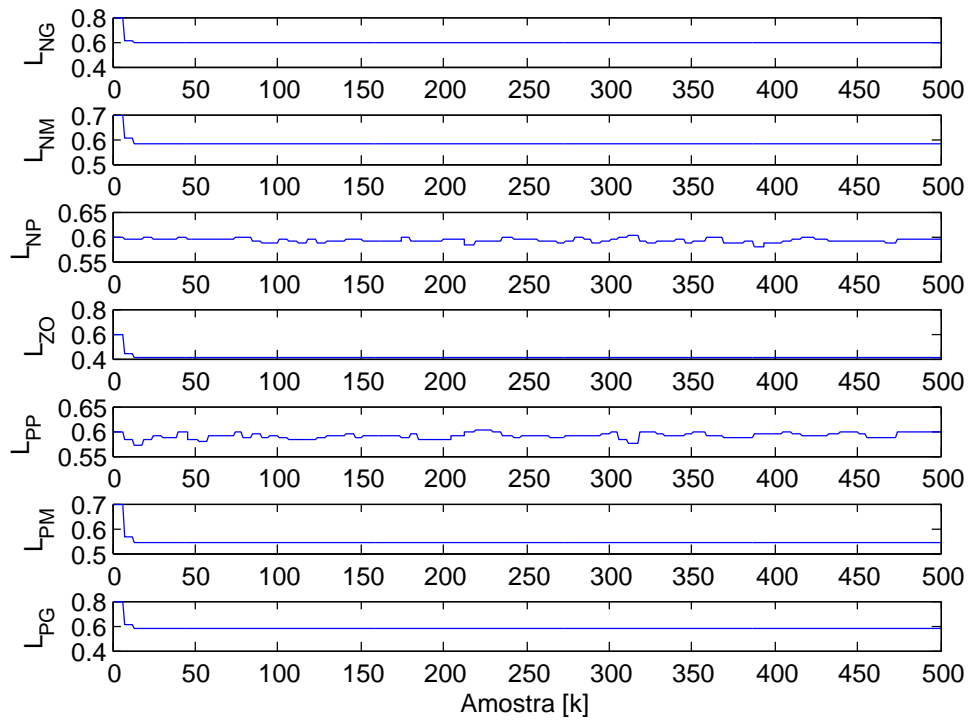


Figura 6.27: Evolução das larguras das funções de pertinência associadas ao Tanque 1, considerando uma configuração em cascata.

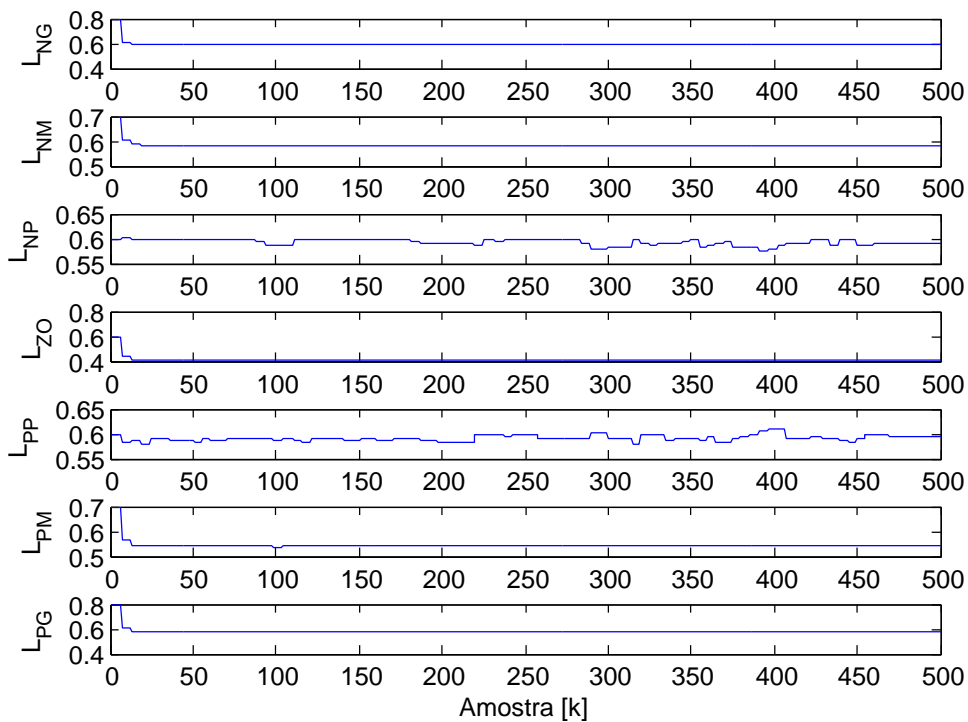


Figura 6.28: Evolução das larguras das funções de pertinência associadas ao Tanque 2, considerando uma configuração em cascata.

6.5 Análise de resultados

Para se poder fazer uma avaliação sobre os resultados obtidos com os diversos controladores testados, recorrer-se-á a duas métricas que expressam o desempenho do controlador no que se refere ao erro e em relação ao incremento da ação de controlo. As métricas referidas são definidas em termos de RMSE e RMSI³, e dadas pelas seguintes equações:

$$RMSE = \sqrt{\frac{\sum_{k=1}^N [y(k) - r(k)]^T [y(k) - r(k)]}{\sum_{k=1}^N r^T(k) r(k)}} \quad (6.13)$$

$$RMSI = \frac{\sqrt{\sum_{k=1}^N \Delta u^T(k) \Delta u(k)}}{N} \quad (6.14)$$

Os valores obtidos para estas métricas, para as experiências realizadas no sistema de três tanques, encontram-se resumidas na tabela 6.8.

Tabela 6.8: Métricas de desempenho.

Implementação	Abordagem	RMSE	RMSI
Em diferido	Fatores de escala otimizados através do AED	0,0572	0,0112
	Fatores de escala otimizados através do método analítico de segunda ordem	0,0751	0,0160
	Fatores de escala e funções de pertinência otimizados através do AED	0,0556	0,0104
	Fatores de escala e funções de pertinência otimizados através do método analítico de segunda ordem	0,0626	0,0106
Em linha	Fatores de escala e funções de pertinência otimizados através do método analítico de segunda ordem	0,0839	0,0125
	Fatores de escala e funções de pertinência otimizados através do método analítico de segunda ordem, com configuração em cascata	0,0847	0,0120

³RMSE=Root Mean Square of Error; RMSI=Root Mean Square of control action Increment

A partir da análise da tabela 6.8 podem ser retiradas várias conclusões, as quais se encontram enumeradas em seguida:

- A otimização em diferido dos fatores de escala em conjunto com as funções de pertinência origina melhores resultados face à otimização em diferido considerando apenas os fatores de escala;
- O AED proporciona um melhor desempenho do controlador PID difuso do que o método analítico de segunda ordem, tanto para a abordagem em que apenas os fatores de escala são otimizados, como para o caso em que os fatores de escala e funções de pertinência são otimizadas simultaneamente. Isto deve-se ao facto do AED ser um método de otimização global, enquanto que o método analítico de segunda ordem apenas assegura um mínimo local;
- Os parâmetros do controlador PID difuso otimizados através das implementações em diferido originam um melhor desempenho relativamente às implementações em-linha, quer seja usada uma configuração de otimização simultânea ou em cascata. Esta situação pode estar relacionada com as restrições em termos de tempo computacional, existentes na implementação em-linha, que origina soluções subótimas para o problema de otimização subjacente.

CONCLUSÕES E PERSPETIVAS FUTURAS

Neste capítulo apresentam-se as conclusões gerais referentes à presente dissertação, assim como algumas ideias que poderão vir a ser desenvolvidas futuramente.

7.1 Conclusões gerais

Esta dissertação abordou o problema da sintonização dos parâmetros do controlador PID difuso do tipo Mamdani, usando o AED (Algoritmo Evolutivo Diferencial). Esta sintonização foi realizada tendo em conta um problema de otimização não linear com restrições, onde é explicitamente considerado o desempenho em malha fechada.

Para além da habitual sintonização dos fatores de escala do controlador PID difuso, nesta dissertação foi também estudado o problema da incorporação de mecanismos de adaptação ao nível das funções de pertença.

A primeira metodologia tida em conta baseou-se na otimização em diferido dos fatores de escala do controlador PID difuso, usando o AED. Relativamente à segunda abordagem, esta considerou a otimização em diferido dos fatores de escala juntamente com as funções de pertença, usando o mesmo algoritmo evolutivo. Estas duas metodologias foram também realizadas recorrendo a um método analítico de segunda ordem. Por fim, o problema de otimização dos fatores de escala e funções de pertença foi implementado através de uma abordagem em-linha, usando um método analítico de segunda ordem, baseada numa arquitetura de computação distribuída.

Os controladores PID difusos otimizados através das diversas metodologias foram testados num sistema de referência, o AMIRA[®] DTS200.

Com base nos resultados obtidos, constatou-se que os controladores PID difusos com fatores de escala e funções de pertinência otimizados em diferido apresentam um melhor desempenho em relação à otimização em diferido considerando apenas os fatores de escala. Verificou-se também que o AED gera melhores resultados do que o método analítico de segunda ordem. Relativamente à abordagem em-linha verificou-se um desempenho inferior ao da implementação em diferido.

Finalmente, no âmbito da dissertação foram publicados três artigos em conferências internacionais:

- A. Sebastião, C. Lucena, L. Palma e P. Gil. "Fuzzy-PID Gains Tuning: Differential Evolution versus Second Order Analytical Algorithms". Em: *10th Asian Control Conference - Malaysia* (2015);
- A. Sebastião, L. Palma, A. Cardoso e P. Gil. "Fuzzy-PID Controller Optimization: A Differential Evolution Approach". Em: *International Conference on Information Processing and Control Engineering - Bauman Moscow State Technical University, Russia* (2015);
- A. Sebastião, C. Lucena, L. Palma, A. Cardoso e P. Gil. "Optimal Tuning of Scaling Factors and Membership functions for Mamdani Type PID Fuzzy Controllers". Em: *International Conference on Control, Automation and Robotics - Singapore* (2015).

e submetidos dois a revistas:

- P. Gil, A. Sebastião, C. Lucena e L. Palma. "Tuning of Mamdani-Type Fuzzy PID Controllers Based on a Constrained Nonlinear Optimisation Approach"; Em: *IEEE Transactions on Cybernetics* (2015);
- P. Gil, A. Sebastião, C. Lucena e L. Palma. "Optimal Tuning of Fuzzy-PID Controllers: An Empirical Study". Em: *IEEE Transaction on Industrial Informatics* (2015).

7.2 Perspetivas futuras

No decorrer do desenvolvimento da dissertação surgiram algumas ideias que poderão vir a ser desenvolvidas futuramente, designadamente:

- Aperfeiçoar o método de otimização global em termos de velocidade de convergência de modo a poder ser usado numa configuração em-linha;
- Comparar o AED com outros algoritmos evolucionários em termos de desempenho e eficiência computacional;
- Adicionar uma ferramenta computacional, como por exemplo uma interface gráfica, ao MATLAB, para apoio ao projeto de controladores.

BIBLIOGRAFIA

- [AOAL08] A. Al-Odienat e A. A. Al-Lawama. "The Advantages of PID Fuzzy Controllers Over The Conventional Types". Em: *American Journal of Applied Sciences* 5 (2008), pp. 653–658.
- [Asc95] C. Aschmann. *Fuzzy systems for information processing*. IOS Press, 1995.
- [BM07] B. Babu e S. Munawar. "Differential evolution strategies for optimal design of shell-and-tube heat exchangers". Em: *Chemical Engineering Science* 62.14 (2007), pp. 3720–3739.
- [BJ08] M. Baczyński e B. Jayaram. *An Introduction to Fuzzy Implications*. Springer, 2008.
- [BT00] P. T. Boggs e J. W. Tolle. "Sequential quadratic programming for large-scale nonlinear optimization". Em: *Journal of Computational and Applied Mathematics* 124.1 (2000), pp. 123–137.
- [Bou+10] H. Boubertakh, M. Tadjine, P.-Y. Glorennec e S. Labiod. "Tuning fuzzy PD and PI controllers using reinforcement learning." Em: *ISA Transactions* 49.4 (2010), pp. 543–551.
- [BM+96] B. Bouchon-Meunier, M. Dotoli e B. Maione. "On the choice of membership functions in a mamdani-type fuzzy controller". Em: Citeaser, 1996.
- [DA11] S. Dasgupta e G. Agnihotri. "The modern Weighted Fuzzy First Maxima Defuzzification Technique for Fuzzy Control of VSCHVDC Converters". Em: *J. Electrical Systems* 7.3 (2011), pp. 332–342.
- [Dha11] M. Dhar. "Fuzzy sets Towards forming Boolean algebra". Em: *International Journal Of Energy, Information and communications, Korea* 2.4 (2011), pp. 137–142.
- [Dri+93] D. Driankov, H. Hellendoorn e M. Reinfrank. *An introduction to fuzzy control*. New York, NY, USA: Springer-Verlag New York, Inc., 1993.
- [Eld92] S. K. Eldersveld. *Large-scale sequential quadratic programming algorithms*. Rel. téc. DTIC Document, 1992.
- [Fen06] G. Feng. "A Survey on Analysis and Design of Model-Based Fuzzy Control Systems". Em: *Fuzzy Systems, IEEE Transactions on* 14.5 (2006), pp. 676–697.

- [FL98] R Fletcher e S Leyffer. "User manual for filterSQP". Em: *Numerical Analysis Report NA/181, Department of Mathematics, University of Dundee, Dundee, Scotland* (1998).
- [Gil+14] P. Gil, C. Lucena, A. Cardoso e L. Palma. "Gains Tuning of Fuzzy PID Controllers for MIMO Systems: A Performance-Driven Approach". Em: *Fuzzy Systems, IEEE Transactions on* pp (2014), p. 1.
- [Gil+05] P. E. Gill, W. Murray e M. A. Saunders. "SNOPT: An SQP Algorithm for Large-Scale Constrained Optimization". Em: *SIAM Review* (2005), pp. 99–131.
- [God00] J. Godjevac. "Comparison between PID and fuzzy control". Em: (2000).
- [Gra07] D. Graupe. *Principles of artificial neural networks*. Vol. 6. World Scientific, 2007.
- [Gri+09] I. Griva, S. G. Nash e A. Sofer. *Linear and nonlinear optimization*. Siam, 2009.
- [HL09] F Herrera e M Lozano. "Fuzzy evolutionary algorithms and genetic fuzzy systems: a positive collaboration between evolutionary algorithms and fuzzy systems". Em: *Computational Intelligence*. Springer, 2009, pp. 83–130.
- [Ilo+03] J. Ilonen, J.-K. Kamarainen e J. Lampinen. "Differential evolution training algorithm for feed-forward neural networks". Em: *Neural Processing Letters* 17.1 (2003), pp. 93–105.
- [Jan07] J. Jantzen. *Foundations of Fuzzy Control*. John Wiley & Sons, Ltd, 2007.
- [JS99] R. Joshi e A. C. Sanderson. "Minimal representation multisensor fusion using differential evolution". Em: *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on* 29.1 (1999), pp. 63–76.
- [Kee11] K. J. Keesman. *System identification: an introduction*. Springer Science & Business Media, 2011.
- [Kis08] N. Kishor. "Nonlinear predictive control to track deviated power of an identified NNARX model of a hydro plant". Em: *Expert Systems with Applications* 35.4 (2008), pp. 1741–1751.
- [Kos92] B. Kosko. "Fuzzy systems as universal approximators". Em: *Proc. IEEE Int Fuzzy Systems Conf* (1992), pp. 1153–1162.
- [Kum+11] V. Kumar, B. Nakra e A. Mittal. "A review of classical and fuzzy PID controllers". Em: *International Journal of Intelligent Control Systems* 16.3 (2011), pp. 170–181.
- [Lee90a] C. C. Lee. "Fuzzy logic in control systems: fuzzy logic controller. II". Em: *Systems, Man and Cybernetics, IEEE Transactions on* 20.2 (1990), pp. 419–435.
- [Lee90b] C. Lee. "Fuzzy logic in control systems: fuzzy logic controller. I". Em: *Systems, Man and Cybernetics, IEEE Transactions on* 20.2 (1990), pp. 404–418.

- [Lee06] D. Lee. "A generalized approach for analyzing transportation user perception using fuzzy sets". Tese de doutoramento. The Pennsylvania State University, 2006.
- [LG96] H.-X. Li e H. Gatland. "Conventional fuzzy control and its enhancement". Em: *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on* 26.15 (1996), pp. 791–797.
- [Liu02] G. Liu. *Nonlinear Identification and Control*. Springer-Verlag New York, Inc., 2002.
- [Lju99] L Ljung. *System identification: theory for the user*. Upper Saddle River: Prentice Hall, 1999.
- [Luc12] C. I. M. d. Lucena. "Aplicação de técnicas de controlo óptimo difuso em ambientes distribuídos". Tese de mestrado. Universidade Nova de Lisboa, 2012.
- [LY08] D. G. Luenberger e Y. Ye. *Linear and nonlinear programming*. Vol. 116. Springer Science & Business Media, 2008.
- [Mal+11] R. Mallipeddi, P. N. Suganthan, Q.-K. Pan e M. F. Tasgetiren. "Differential evolution algorithm with ensemble of parameters and mutation strategies". Em: *Applied Soft Computing* 11.2 (2011), pp. 1679–1696.
- [Mis+96] D. Misir, H. A. Malki e G. Chen. "A heuristic approach to determine the gains of a fuzzy pid controller". Em: *Proceedings of the 1996 ACM symposium on Applied Computing*. ACM. 1996, pp. 609–613.
- [MM97] G. Mouzouris e J. Mendel. "Dynamic non-Singleton fuzzy logic systems for nonlinear modeling". Em: *Fuzzy Systems, IEEE Transactions on* 5.2 (1997), pp. 199 –208.
- [Nov95] V. Novák. "Linguistically oriented fuzzy logic control and its design". Em: *International Journal of Approximate Reasoning* 12.3 (1995), pp. 263–277.
- [PP10] S. K. Pal e J. F. Peters. *Rough Fuzzy Image Analysis: Foundations and Methodologies*. CRC Press, 2010.
- [Pis+11] H. N. Pishkenari, S. Mahboobi e A. Alasty. "Optimum synthesis of fuzzy logic controller for trajectory tracking by differential evolution". Em: *Scientia Iranica* 18.2 (2011), pp. 261–267.
- [Rah+08] M. H. F. Rahiman, M. N. Taib e Y. M. Salleh. "Assessment of NNARX structure as a global model for self-refilling steam distillation essential oil extraction system". Em: *Information Technology, 2008. ITSIM 2008. International Symposium on*. Vol. 3. IEEE. 2008, pp. 1–7.
- [Ran04] A. Ranganathan. "The levenberg-marquardt algorithm". Em: *Tutorial on LM Algorithm* (2004), pp. 1–5.

- [Rez+00] L. Reznik, O. Ghanayem e A. Bourmistrov. "PID plus fuzzy controller structures as a design base for industrial applications". Em: *Engineering applications of artificial intelligence* 13.4 (2000), pp. 419–430.
- [Rib04] M. I. Ribeiro. "Kalman and Extended Kalman Filters: Concept, Derivation and Properties". Em: *Institute for Systems and Robotics* (2004), p. 43.
- [Riz+11] P. Rizol, L. Mesquita e O. Saotome. "Lógica Fuzzy tipo-2". Em: *Revista SODEBRAS* 6 (2011), pp. 27–46.
- [Ros09] T. J. Ross. *Fuzzy logic with engineering applications*. John Wiley & Sons, 2009.
- [Sar10] S. P. Saramago. "Técnicas Heurísticas de Otimização Aplicadas em Engenharia". Em: *Horizonte Científico* 3.2 (2010).
- [SSJ+08] S. P. Saramago, V. Steffen Jr et al. "Introdução às Técnicas de Otimização em Engenharia". Em: *Horizonte científico* 1.9 (2008).
- [Sjö+95] J. Sjöberg, Q. Zhang, L. Ljung, A. Benveniste, B. Delyon, P.-Y. Glorennec, H. Hjalmarsson e A. Juditsky. "Nonlinear black-box modeling in system identification: a unified overview". Em: *Automatica* 31.12 (1995), pp. 1691–1724.
- [Sto96] R. Storn. "Differential evolution design of an IIR-filter". Em: *Evolutionary Computation, 1996., Proceedings of IEEE International Conference on*. IEEE. 1996, pp. 268–273.
- [SP95] R. Storn e K. Price. "Differential evolution-a simple and efficient adaptive scheme for global optimization over continuous spaces". Em: 3 (1995).
- [Wan99] L.-X. Wang. *A course in fuzzy systems*. Prentice-Hall press, USA, 1999.
- [Wil63] R. B. Wilson. "A Simplicial Method for Convex Programming". Ph.D. thesis. Harvard University, 1963.
- [Zad65] L. Zadeh. "Fuzzy sets". Em: *Information and control* 8.3 (1965), pp. 338–353.
- [Zad75] L. Zadeh. "The concept of a linguistic variable and its application to approximate reasoning I". Em: *Information Sciences* 8.3 (1975), pp. 199–249.
- [Zad73] L. A. Zadeh. "Outline of a new approach to the analysis of complex systems and decision processes". Em: *Systems, Man and Cybernetics, IEEE Transactions on* 1 (1973), pp. 28–44.
- [Zad08] L. A. Zadeh. "Is there a need for fuzzy logic?" Em: *Information Sciences* (2008).
- [Zhu05] Z. Zhu. "An efficient sequential quadratic programming algorithm for nonlinear programming". Em: *Journal of Computational and Applied Mathematics* 175.2 (2005), pp. 447–464.
- [Zon06] Z. Zong. *Information-Theoretic Methods for Estimating of Complicated Probability Distributions*. Vol. 207. Elsevier, 2006.